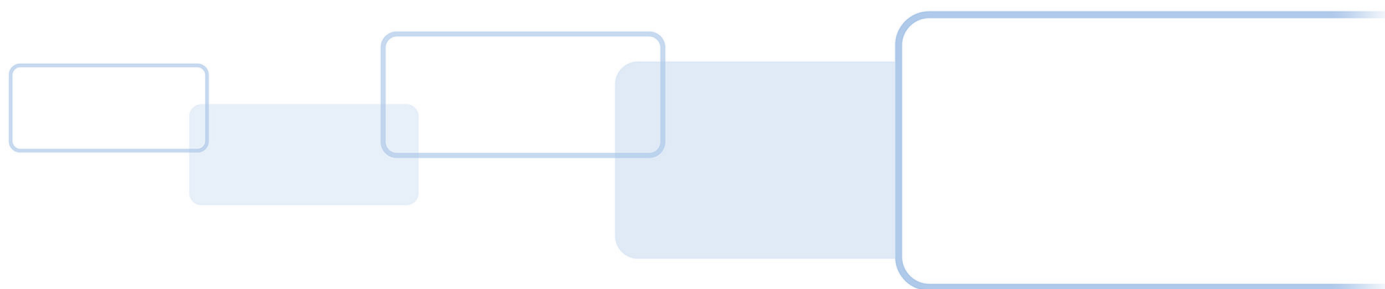




# **OMNIKEY® 5x27CK KEYBOARD WEDGE CONFIGURATION AND CUSTOM REPORT USER GUIDE**

5127-902, Rev E.3  
November 2019



## Copyright

© 2011 - 2019 HID Global Corporation/ASSA ABLOY AB. All rights reserved.

This document may not be reproduced, disseminated or republished in any form without the prior written permission of HID Global Corporation.

## Trademarks

HID Global, HID, the HID Brick logo, the Chain Design, HID Mobile Access, Indala, 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/EV2, and MIFARE Ultralight are registered trademarks of NXP B.V. and are used under license.

## Revision History

Date	Description	Version
November 2019	Added OK5127CK Reader Core information. Added note at end of section 2.2.5. Added extra configuration examples in section 6.	E.3
January 2019	Added information to support Service Pack 2 (FW 01.02.00f7)	E.2
January 2018	Added information on OK5425 Gen2 and OK5127 Mini SP1	E.1
March 2016	Added information on OK5127CK-Mini.	E.0
December 2014	Extra detail added to tech order setting.	D.3

## Contacts

For additional offices around the world, see [www.hidglobal.com/contact/corporate-offices](http://www.hidglobal.com/contact/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](http://www.hidglobal.com/support)



# Contents

<b>Chapter 1: Overview</b>	<b>7</b>
1.1 References	7
1.2 Abbreviations and definitions	8
1.3 Firmware version information	8
1.4 Supported RFID technologies	9
1.4.1 LF technologies (125 kHz)	9
1.4.2 HF technologies (13.56 MHz)	9
1.4.3 Bluetooth support (OK5127CK-Mini, OK5427 Gen 2, and OK5127 Reader Core)	10
1.5 Modes of operation	10
1.5.1 Ethernet Emulation Mode (EEM)	10
1.5.2 CCID	11
1.5.3 Keyboard Wedge	12
1.5.4 Custom Report	12
<b>Chapter 2: Reader web-based management tool interface</b>	<b>13</b>
2.1 Setup	13
2.1.1 Ethernet Emulation Mode	13
2.1.2 Web browser	14
2.2 Navigating the Reader Management Tool	14
2.2.1 Accessing the web interface	14
2.2.2 Navigating the tabs	15
2.2.3 Changing settings	16
2.2.4 Downloading and uploading configurations	17
2.2.5 Setting a web server password	19
2.3 Card type processing priority	20
2.4 Polling configuration	21
<b>Chapter 3: Keyboard wedge mode</b>	<b>23</b>
3.1 Card In event	24
3.1.1 Card Out event	24
3.2 Navigating the Keyboard Wedge configuration tabs	25
3.3 General Config tab	26
3.3.1 KBW Enable options	26
3.3.2 Global keystroke events	27
3.3.3 Keyboard options	28
3.3.4 Keyboard wedge encryption	29
3.4 Card Data Selection tab	30

- 3.4.1 Configure data fields for each card type..... 31
- 3.5 Data manipulation ..... 32
- 3.6 Input Data Manipulation tab ..... 33
  - 3.6.1 PACS Leading Byte..... 33
  - 3.6.2 Binary Reverse..... 33
  - 3.6.3 Byte Reverse ..... 34
  - 3.6.4 Bit Padding..... 35
  - 3.6.5 Logic Operations..... 35
  - 3.6.6 Out 1 Data Manipulation tab ..... 36
    - 3.6.6.1 String Format ..... 36
    - 3.6.6.2 String Filtering..... 36
    - 3.6.6.3 String Truncating ..... 37
  - 3.6.7 Out 2 Data Manipulation tab..... 38
    - 3.6.7.1 String Padding..... 38
    - 3.6.7.2 Pre- and Post-strokes ..... 38
- 3.7 Supported keystroke & command characters ..... 39
  - 3.7.1 Supported printable characters ..... 39
  - 3.7.2 Pre- and post-stroke supported control characters ..... 39
  - 3.7.3 Extended ASCII Character Set (OK5427 Gen2/OK5127CK Mini/  
OK5127CK Reader Core onwards)..... 42
  - 3.7.4 Reader command keystrokes (controlling reader behavior)..... 42
    - 3.7.4.1 [PAUSE xxx] ..... 42
    - 3.7.4.2 [LED\_BUZZ] ..... 43
- 3.8 Secure messaging on MIFARE DESFire EV2 ..... 43
- 3.9 Keyboard Wedge output via UART ..... 43
- 3.10 Maximum output size ..... 44
- Chapter 4: Custom Report mode ..... 45**
  - 4.1 Example Custom Report output across USB HID interface ..... 45
    - 4.1.1 Output example: ..... 47
- Chapter 5: Additional settings ..... 49**
  - 5.1 LEDs & buzzer ..... 49
    - 5.1.1 Navigating the LEDs & Buzzer tab ..... 49
    - 5.1.2 Legacy keyboard wedge LED & buzzer behavior ..... 49
    - 5.1.3 Configuring the LED and buzzer behavior..... 50
      - 5.1.3.1 Incorrect LED/buzzer sequence ..... 51
      - 5.1.3.2 Incorrect zero duration ..... 51
  - 5.2 Host interfaces ..... 51
  - 5.3 Navigating the Host Interfaces tab ..... 52
    - 5.3.1 EEM IP interface parameters..... 52
    - 5.3.2 USB Interface Parameters..... 53
- Chapter 6: OMNIKEY® 5x27 configuration examples ..... 55**
  - 6.1 Example 1 - Reading iCLASS® card PACS data ..... 55

6.2	Example 2 - Reading MIFARE card CSN . . . . .	56
6.3	Example 3 - HID iCLASS PACS data filtering . . . . .	57
6.4	Example 4 - Prox card string padding . . . . .	59
6.5	Example 5 - HID iCLASS, standard 26 bit, FC and CN . . . . .	60
6.6	Example 6 - PIV 75 bit card number . . . . .	62
6.7	HID PROX 26-bit format H10301 facility code and user ID (decimal output) . . . . .	63
6.8	Indala® PROX default format . . . . .	65
6.9	iCLASS H10304 format facility code and user ID (decimal output) . . . . .	67
6.10	MIFARE Classic 26-bit format facility code and user ID (decimal output) . . . . .	70
6.11	MIFARE Classic sector read, including load keys: . . . . .	72
6.12	MIFARE DESFire H10302 format, user ID (decimal output): . . . . .	74
6.13	MIFARE Ultralight sector read, including load keys . . . . .	77
6.14	FeliCa CSN with HEX and DEC output: . . . . .	78
6.15	MIFARE DESFire custom application read and loading keys: . . . . .	79
6.16	Seos credentials, corporate 1000 format, facility code and user ID (decimal output) . . . . .	80
6.17	MIFARE Plus custom sector read with load keys . . . . .	82
<b>Appendix A: Description of fields . . . . .</b>		<b>85</b>
A.1	Enable Card Type . . . . .	85
A.2	Card In Event Keystrokes . . . . .	85
A.3	Pre-strokes . . . . .	85
A.4	Post-strokes . . . . .	85
A.5	CSN . . . . .	86
A.6	CSN Custom . . . . .	86
A.6.1	Reverse . . . . .	86
A.6.2	Offset . . . . .	86
A.6.3	Length . . . . .	86
A.7	PACS . . . . .	86
A.8	PACS Custom . . . . .	86
A.8.1	Offset . . . . .	86
A.8.2	Length . . . . .	86
A.9	iCLASS Custom Fields . . . . .	87
A.9.1	Key . . . . .	87
A.9.2	Key Type . . . . .	87
A.9.3	Book . . . . .	87
A.9.4	Page . . . . .	87
A.9.5	Block . . . . .	87
A.9.6	Offset . . . . .	87
A.9.7	Length . . . . .	87
A.10	MIFARE Classic and MIFARE Plus Custom Fields . . . . .	88

A.10.1	Key .....	88
A.10.2	Key Type .....	88
A.10.3	Sector .....	88
A.10.4	Block .....	88
A.10.5	Offset .....	88
A.10.6	Length .....	88
A.11	MIFARE Ultralight Custom Fields .....	89
A.11.1	Key .....	89
A.11.2	Page .....	89
A.11.3	Offset .....	89
A.11.4	Length .....	89
A.12	MIFARE DESFire and MIFARE DESFire EV1 Custom Fields .....	90
A.12.1	App ID .....	90
A.12.2	File Num. ....	90
A.12.3	Offset .....	90
A.12.4	Length .....	90
A.12.5	Card Key .....	90
A.12.6	Rdr Key .....	90
A.12.7	Auth .....	90
A.12.8	File Type .....	90
A.12.9	File Comms .....	91
A.13	MIFARE DESFire EV1 and MIFARE DESFire EV2 Custom Fields .....	91
A.13.1	Start .....	91
A.13.2	Len .....	91
A.13.3	Encryption .....	91
A.13.4	AV1 Diversify (MIFARE DESFire EV1 only) .....	91
A.13.5	CT value .....	91
A.14	PIV Specific Fields .....	92
A.14.1	FASC-N .....	92
A.14.2	GUID .....	92
A.14.3	75-Bit GSA .....	92
A.14.4	FASC-N Custom .....	92
A.14.5	FASC-N Custom Remove Parity .....	92
A.14.6	FASC-N Reverse BCN .....	92
A.15	CEPAS Custom Fields .....	92
A.15.1	CAN .....	92
<b>Appendix B: Extended ASCII character set .....</b>		<b>93</b>



# Chapter 1

## Overview

---

HID Global's OMNIKEY® 5x27CK readers open new market opportunities for system integrators seeking simple integration and development of readers using the standard Circuit Card Interface Device (CCID).

With the keyboard wedge functionality, users of OMNIKEY 5x27 CK readers can retrieve data from a card that is presented to the reader and directly input the card data into an application using keystroke emulation. This eliminates the need for customers to manually enter the card data into an application.

This guide explains how to setup the reader to use different card types in the Keyboard Wedge mode using the web browser interface.

To use the reader browser interface, the EEM-USB driver must be installed. For installation instructions, see the *OMNIKEY 5x27CK Quick Start Guide* (5127-901).

**Note:** HID provides various Service Packs for the OMNIKEY 5x27CK. Some functions have been introduced with later Service Packs only. These exceptions are noted in this user guide. For downloading the latest Service Pack for your OMNIKEY 5x27CK reader, access the Developer Center: [www.hidglobal.com/developer-center/omnikey-5x27ck](http://www.hidglobal.com/developer-center/omnikey-5x27ck).

Service Packs are available in the **Downloads** section, which requires a user account. Check the firmware version of the OMNIKEY 5x27CK Reader from the **General Overview** tab in the built-in web interface. See *Section 2: Reader web-based management tool interface*.

### 1.1 References

Document Number	Description
5127-901	Quick Start Guide
5127-903	Software Developer Guide
AN0407	Firmware Upgrade

## 1.2 Abbreviations and definitions

The following acronyms and abbreviations may be used in this document:

Abbreviation	Description
ASK	Amplitude Shift Key - a modulation schema for RF communications
BLE	Bluetooth Low Energy
CCID	Chip Card Interface Device Protocol
CHUID	Card Holder Unique Identifier
Config	Short for "Configuration"
CSN	Chip Serial Number or Card Serial Number
EEM	Ethernet Emulation Mode
FSK	Frequency Shift Key - a modulation schema for RF communications
FW	Firmware
GUID	Global Unique Identifier
HF	High Frequency - 13.56 MHz
HTTP	Hyper Text Transfer Protocol
HW	Hardware
KBW	Keyboard Wedge
LF	Low Frequency - 125 kHz "Prox"
OS	Operating System
PACS	Physical Access Control System
PSK	Phase Shift Key -a modulation schema for RF communications
RCN	Random Chip Number
RFID	Radio Frequency Identification

## 1.3 Firmware version information

There are two generations of OMNIKEY 5x27: Gen1 and Gen2. The firmware versions for both generations of readers have independent numbering. The table below helps with identifying actual version of the reader and firmware version.

Part Number (starts with)	Generation	Firmware versions
R5x270001	Gen1	01000000 - 04020500
R54270101	Gen2	01.02.00f7
R54270111		
R51270010		
R51270020		
OK5127CK Reader Core		

In this document, to ease the ability to recognize the firmware version, numbering for Gen2 readers is denoted with dots (e.g. 01.00.0019, where for Gen1 it would be 01000019).



## 1.4 Supported RFID technologies

### 1.4.1 LF technologies (125 kHz)

Card Type	Firmware Version	Data Availability	Protocol Polling <sup>1</sup>
FSK Prox (e.g. HID Prox, AWID)	01000000 or higher	PACS	Prox
PSK Prox (e.g. Indala®)	03000000 or higher		
ESK Prox (e.g. EM4450, Acura)			
HITAG 1, 2 and S	01.00.0069 or higher		

1. The Polling Config tab is found under the **Contactless Config** tab.

### 1.4.2 HF technologies (13.56 MHz)

Card Type	Firmware Version	Data Availability	Protocol Polling <sup>1</sup>
Seos®	03000000 or higher	CSN, PACS, Custom	ISO 14443A
HID iCLASS®	01000000 or higher	CSN, PACS, Custom	iCLASS 15693
MIFARE Classic			
MIFARE Ultralight / C		CSN, Custom	
MIFARE DESFire			
MIFARE DESFire EV1 <sup>2</sup>			CSN, PACS, Custom
MIFARE DESFire EV2	SP2 FW version (when known)		
MIFARE Plus <sup>3</sup>	01000000 or higher	CSN, Custom	
FeliCa	01.00.0069 or higher	CSN, Custom	FeliCa
PIV	04000000 or higher	CSN, FASC-N, GUID, GSA, Custom	ISO 14443A & B
CEPAS	SP2 FW version (when known)	CSN, CAN, Custom	ISO 14443B
Generic ISO14443A	04000000 or higher	CSN, PACS, Custom	ISO 14443A
Generic ISO14443B			ISO 14443B
Generic ISO15693			iCLASS 15693

1. The **Polling Config** tab is found under the **Contactless Config** tab.

2. MIFARE DESFire EV1 (MAC secured, DES/3DES, 3K3DES and AES encrypted - firmware 02000000 or higher; diversification - firmware 04000000 or higher).

3. Security Level 3 requires firmware 04000000 or higher.

### 1.4.3 Bluetooth support (OK5127CK-Mini, OK5427 Gen 2, and OK5127 Reader Core)

HID Seos credentials can be read from any phone which satisfies the following requirements:

- Either Android version 4.3 or later or iOS 7 or later
- Bluetooth 4.0
- HID Mobile Access® app installed and running

This feature is available only on the OK5127CK-Mini, OK5127 Reader Core, and OMNIKEY 5427CK Gen2 readers. It is not available on the original OK5127CK or OK5427CK.

## 1.5 Modes of operation

Due to the way that some operating systems handle USB devices, HID suggests that anyone using KBW or Custom Report mode designates two OMNIKEY 5x27 units for use with their PC to enable the following work flow:

- OMNIKEY 5x27 in KBW Mode - all testing and setup of parameters
- OMNIKEY 5x27 in CCID Mode - programming configuration cards
- Apply all KBW and Custom Report Mode Settings via configuration card

If this approach is not followed, the computer user must carefully manage the instances of the devices to prevent registry corruption.

### 1.5.1 Ethernet Emulation Mode (EEM)

EEM is enabled by default to manage configuration settings via the embedded web based management tool or over TFTP. EEM operates in addition to any other interface to allow for access to configuration settings.

The only way to recover EEM once disabled is via a configuration card, MIB command in CCID Mode or by SetFeature request.

To switch on CCID mode (when Keyboard Wedge mode is active), send a HID (Human Interface Device) Set Feature Report request with report ID 0x00 and two bytes: 0xA5, 0x5A. The device will then re-enumerate and communication via CCID will be possible.

Using Set Feature Report request, it is also possible to switch on/off EEM. To enable EEM, send a request with report ID 0x00 and bytes 0xA5 and 0xEE. To disable EEM, use 0xA5, 0xE0.

It is also possible to switch on/off the EEM interface using the HID OMNIKEY Workbench tool, which can be found on the HID Global web site: <https://www.hidglobal.com/drivers>.

#### Enumeration

When EEM is operational, the OMNIKEY 5x27 will enumerate with the OS as a Network Adapter in addition to enumerating as a Smart Card Reader, Keyboard, or Composite USB device. In a Windows environment, the device shown in device manager as **HID USB CDC EEM Ethernet Adapter #n** (where n is the number of occurrence of the device)

The PID/VID for the device in this mode or operation mirrors the PID/VID for the CCID, Keyboard, or custom mode.

### CCID Mode operational



HID USB CDC EEM Ethernet Adapter (5427) #2

Property

Hardware IDs

Value

USB\VID\_076B&PID\_5427&REV\_0100&MI\_01  
USB\VID\_076B&PID\_5427&MI\_01

### Keyboard Wedge or Custom Report mode is operational



HID USB CDC EEM Ethernet Adapter (5428) #2

Property

Hardware IDs

Value

USB\VID\_076B&PID\_5428&REV\_0100&MI\_01  
USB\VID\_076B&PID\_5428&MI\_01

## 1.5.2 CCID

CCID is mainly used for read/write applications or with hosts that cannot support a keyboard input. CCID required an intelligent host and operates as a transparent PC/SC - CCID reader where the host controls every aspect of the card communication.

CCID mode must be active in order to create an OMNIKEY 5x27 configuration card as this requires read/write capability.

CCID mode cannot be operational when Keyboard Wedge mode is operational.

### Enumeration

In CCID mode, the OMNIKEY 5427 enumerates with the OS as a Smart Card Reader.



HID OMNIKEY 5427CK

Property

Hardware IDs

Value

USB\VID\_076B&PID\_5427&REV\_0100&MI\_00  
USB\VID\_076B&PID\_5427&MI\_00

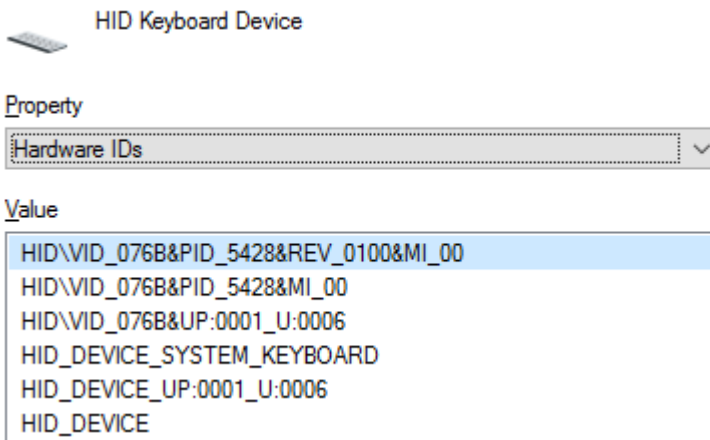
### 1.5.3 Keyboard Wedge

KBW mode supports read only applications and is fully configurable via the built-in web based management tool, TFTP and configuration cards.

In KBW mode, the reader will access, buffer, process and report data as series of keyboard keystrokes to the host as configured.

#### Enumeration

When operating in KBW mode, the OMNIKEY 5x27 enumerates with the OS as a keyboard device.



### 1.5.4 Custom Report

Custom Report mode requires that KBW is enabled within the reader, and outputs the configured data as raw HEX and not keyboard keystrokes.

#### Enumeration

In Custom Report mode the OMNIKEY 5x27 enumerates with the OS as a USB Composite Device in addition to enumerating as a keyboard.



## Reader web-based management tool interface

---

The OMNIKEY® 5x27CK Reader has a built in, web based management tool that can be used to configure many aspects of the reader performance and behavior. This section provides a brief explanation of all the tabs, and the basic functions found under each tab for easy navigation and use.

**Note:** Due to how the Windows OS manages instances of devices, HID recommends that a single 5427CK device is used to build configurations. The configurations should be applied via configuration cards on a different host OS device. If this cannot be done, care must be taken to manage the device instances in Windows to prevent computer issues.

### 2.1 Setup

The web based management tool is intended to allow users to configure device operating parameters manually with an intuitive UI that is easy to understand.

The Web Based UI is simply a user friendly interface which sends commands over to the reader over the EEM HTTP channel. The commands it uses are all documented in the *OMNIKEY 5X27CK Software Developer Guide* (5127-903).

HID suggests all integrators implement configuration/firmware upgrade capability.

When using CCID, it is strongly suggested that you investigate the Abstraction layer command `ProcessKeyboardWedge`, documented in the *OMNIKEY 5X27CK Software Developer Guide* (5127-903). This will often greatly simplify your application, as it transfers much of the sequential process of reading specific data from cards to the reader. It also makes testing and debugging easier.

#### 2.1.1 Ethernet Emulation Mode

The OMNIKEY 5x27 EEM Driver must be downloaded onto the Windows based PC and installed before plugging the reader into the USB port. The EEM Driver can be found on the OMNIKEY 5x27 Developer Center under Downloads, or at <http://www.hidglobal.com/drivers>.

The EEM Driver currently supports the following 32 and 64-bit Windows OS versions:

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7

## 2.1.2 Web browser

As with any web based application, the internet browser directly affects the user experience. HID Global does everything possible to minimize the impact that different web browsers have on the user experience. However, with frequent changes and the fact that the tool is an embedded firmware web based tool, HID Global cannot fully guarantee interoperability with all web browsers.

### Supported web browsers (English versions only)

- Internet Explorer, version 11 (Compatibility Mode must be disabled)
- Firefox, from version 53
- Chrome, from version 58
- Opera, from version 45

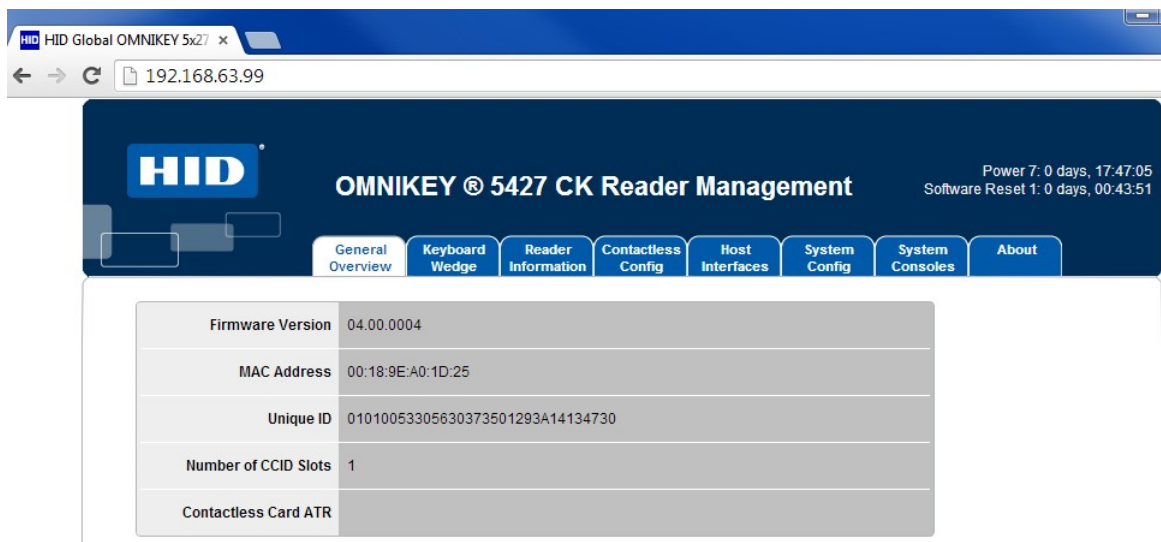
Known issues may exist with different firmware revisions of the OMNIKEY 5x27 and specific browsers. Refer to the firmware release notes for any known issues.

## 2.2 Navigating the Reader Management Tool

**Note:** The PC used to access the web interface must be prepared as described in *Section 2.1: Setup*, then connected to the reader.

### 2.2.1 Accessing the web interface

1. Start a supported web browser.
2. Enter **http://192.168.63.99/** into the address bar and press **Enter**. The OMNIKEY 5x27 web server page launches with the **General Overview** tab selected, which is similar to the following page.



## 2.2.2 Navigating the tabs

The following table describes the functions of each tab.



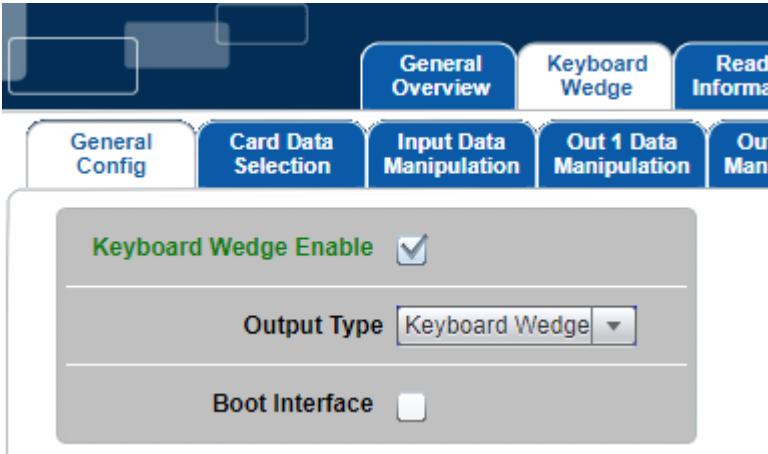
Tab	Description	Intended User Level
<b>General Overview</b>	A quick overview of reader information to include Main Firmware Version, MAC Address, UID of the reader, No. of CCID slots and the Contactless Card ATR.	Novice
<b>Keyboard Wedge</b>	Keyboard Wedge Setup Parameters.	Novice
<b>Reader Information</b>	Full view of the reader firmware and hardware state.	Novice
<b>Contactless Config</b>	RF, BLE and LED/Buzzer register settings.	Novice
<b>Host Interfaces</b>	Host interface configuration items for USB and Ethernet Emulation Mode.	Advanced
<b>System Config</b>	Reader configuration and firmware management to include: <ul style="list-style-type: none"> <li>■ Apply, Reset and Store configuration changes</li> <li>■ Reset all configuration to factory default</li> <li>■ Load and download complete configuration files</li> <li>■ Manage firmware</li> <li>■ Change access levels with passwords</li> </ul>	Firmware and Configuration Parameters: Novice  Change of access levels: Advanced
<b>System Consoles</b>	Interface to view actual USB traffic	Advanced
<b>About</b>	Acknowledgments and legal statements	N/A

### 2.2.3 Changing settings

Modified settings are green at first and turn black when finalized using the **Apply Changes** option specified in step 3.

#### Modify Settings

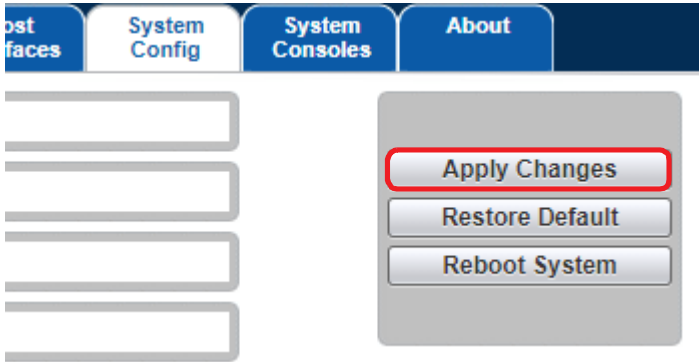
1. Change the configuration parameters as needed. The description or value color changes to green.



2. Press **Enter** to finalize text field changes including special characters such as [ENTER].



3. Navigate to the **System Config** tab and click **Apply Changes**. The changed configuration parameters revert to black



**Note:** The **Reboot System** button is only necessary when changing operational modes (CCID, Keyboard Wedge, Custom Report) so the reader can re-enumerate with the host system as the proper device, and when changing configuration parameters under the Host Interface tab (Changing Ethernet Settings, Enabling UART, etc.). Click **Reboot System** only after clicking **Apply Changes**.

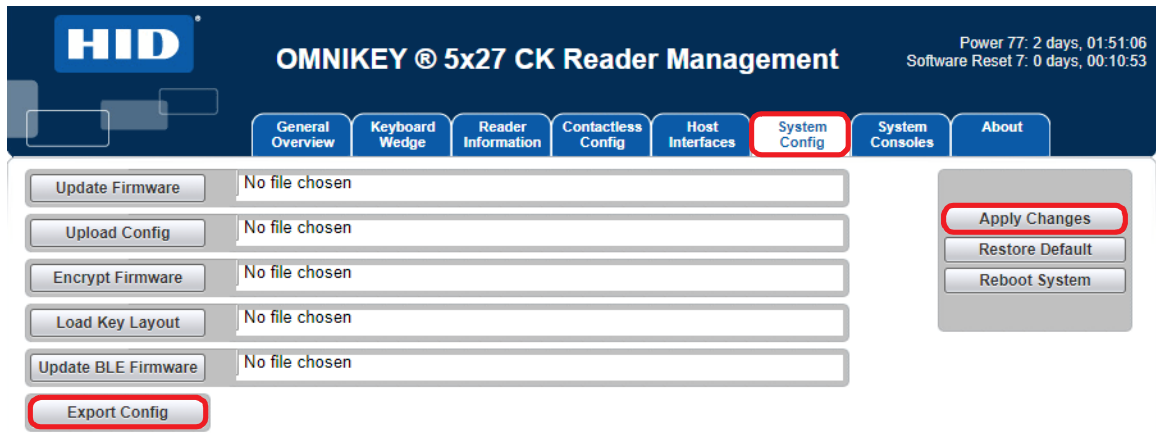


## 2.2.4 Downloading and uploading configurations

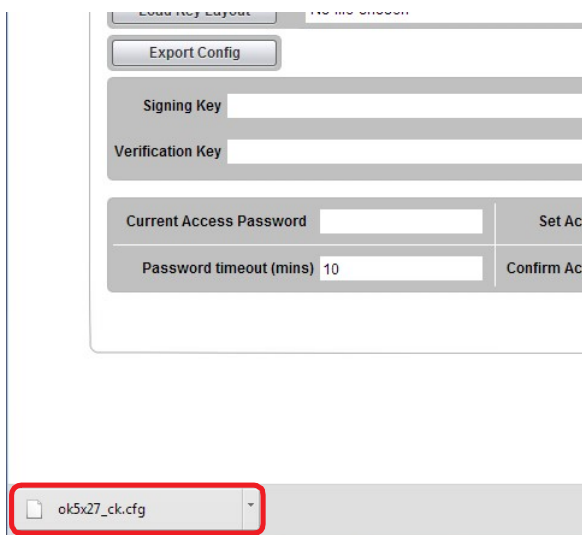
Downloading and uploading configuration files is an important feature of the OMNIKEY 5x27. Once a configuration is fully tested, it can be downloaded and used to make a configuration card using the `hid_ok5x27ck_configcard_tool`, that can be downloaded from the Developer Center.

### Download a configuration file

1. On the **System Config** tab, change all configuration settings as needed.
2. Click **Apply Changes**.
3. Click **Export Config**

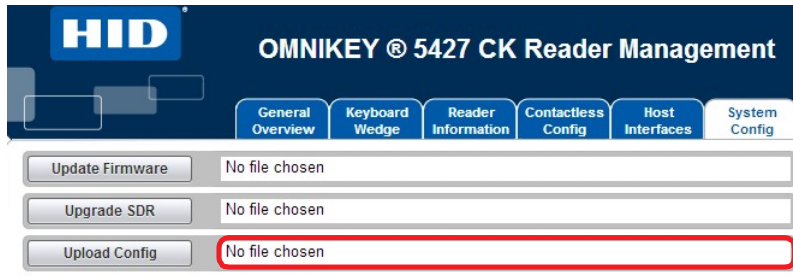


4. Rename the file to be specific to the configuration for future reference (the file will always be named **ok5x27ck.cfg** upon download).

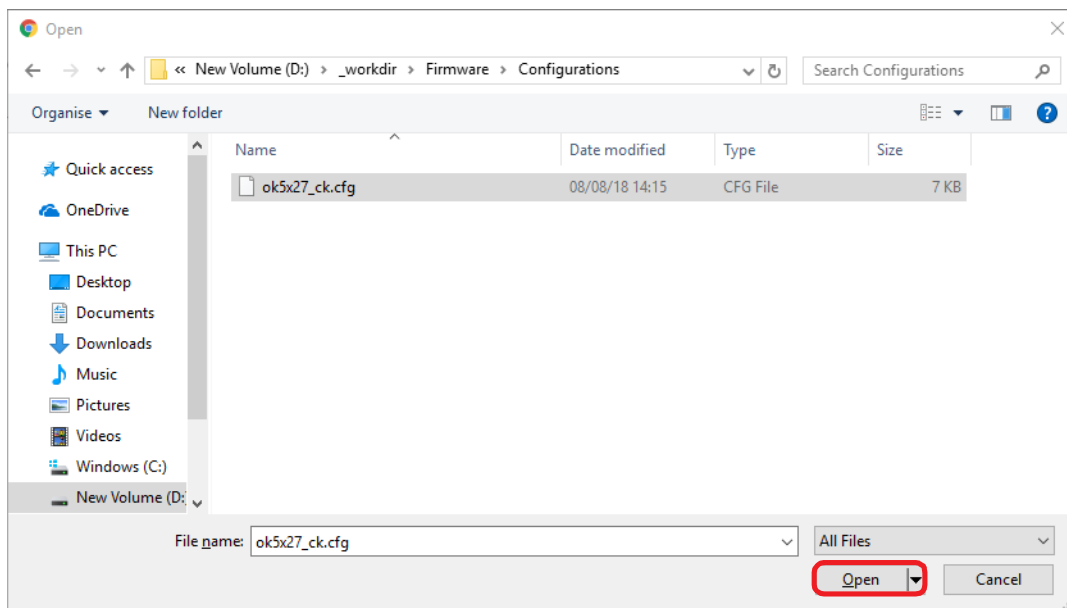


### Upload a configuration file

1. Select a file by clicking in the text box next to the **Upload Config** button.



2. Search for the configuration file in Windows Explorer, select the file and click **Open**.



The configuration file name is displayed in the text box.

3. To upload and apply the configuration contained in the file, click **Upload Config**.



**Note:** Please bear in mind that a MIFARE DESFire Configuration card will update only the parameters available in the web server UI, and does not load keys, change Indala® format, etc. Please contact a HID Sales, Presales Engineer or Field Applications Engineer for more details.

## 2.2.5 Setting a web server password

The web page management tool for the OK5x27CK can be protected by a password. This restricts access to the web page based management tool only.

### Password entry options

To set the password, enter the existing access password, the new password, and confirmation of the new password in the password section of the **System Config** tab.

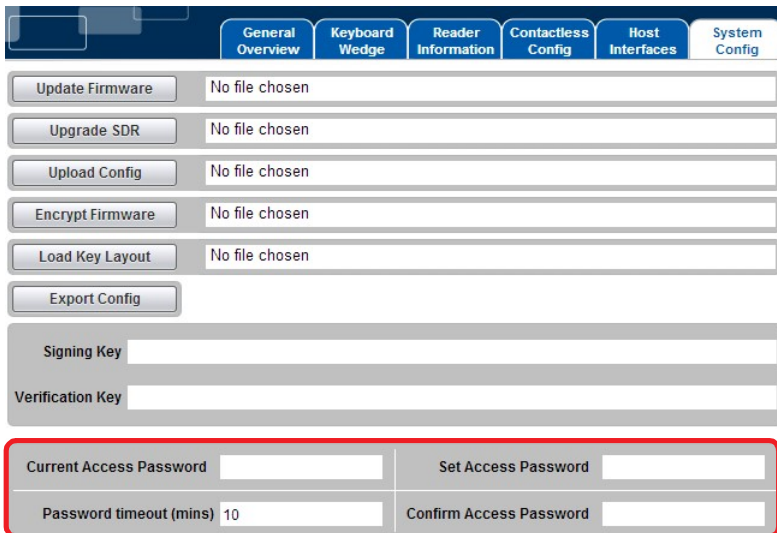
To send the password to the reader, place the cursor in one of the three password fields, then press **Enter**.

If there is no password currently set, leave the **Current Access Password** field blank.

To disable the password, leave both the **Set Access Password** and **Confirm Access Password** fields blank.

Once the password has been sent to the reader it will be necessary to click **Apply Changes** in order for the password to be kept after a system reboot.

The **Password timeout (mins)** field specifies the amount of time in minutes the current login session will last before the user will have to re-enter the password. To use an infinite timeout enter a value of zero (0).



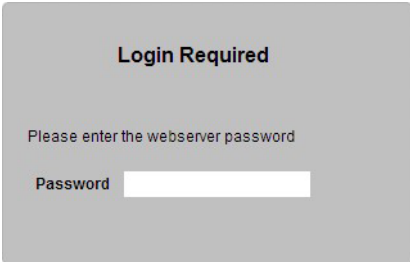
The screenshot shows the 'System Config' tab with several sections. The bottom section, highlighted with a red box, contains four input fields: 'Current Access Password', 'Set Access Password', 'Password timeout (mins)' (with a value of 10), and 'Confirm Access Password'.

If you prefer, this can also be done by sending the following APDU to the reader:

CLA	INS	P1	P2	Lc	Data	
0xFF (Pseudo-APDU)	0x68 (OK5x27CK Command)	0x00 (MIB Command)	0x01 (MIB Control)	Length of password +2	0x05 (Password Entry Command)	ASCII Password + null terminating character

### Login screen

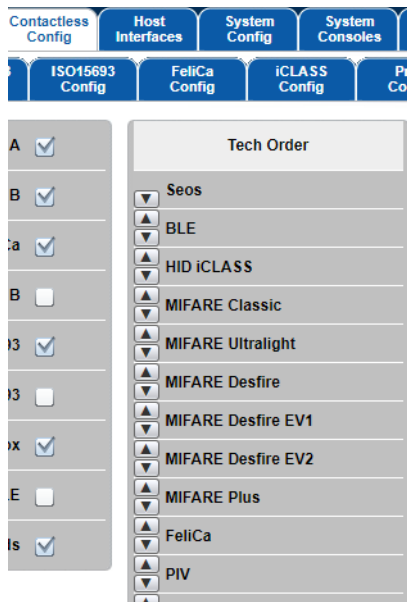
Once a password is set, you will be automatically be presented with a log in screen on accessing the webserver. To login, enter the password created previously. If the password is entered incorrectly there will be a delay of several seconds before the password can be entered again.



**Note:** To recover the password, please contact Technical Support.

## 2.3 Card type processing priority

Card type processing priority makes it possible to reduce the response time for the application to respond to a card presented to the reader. HID recommends that the card processing prioritization is configured for each installation of a device to ensure that the primary card type has priority. To configure the card processing priority, go to the **Contactless Config** tab and use the **Tech Order** arrow buttons as shown below.



**Note:** If Other ISOxxx is configured as the highest priority, the only output reported will be the CSN of the smartcard.

**Note:** It is best practice to place at the top of the priority list the card type that is the primary card at the installation. This will reduce the processing time for the card type and associated data.

## 2.4 Polling configuration

**Note:** In Service Pack 2.0, an additional **Enhanced Polling** option was added. This option influences the Tech Order in the reader (turns it on/off). By default, this option is not selected. This results in the Tech Order being deactivated, which causes dual technology cards to become unsupported. This option increases reader performance by shortening card activation time. If dual card support is needed, the **Enhanced Polling** option must be selected.

This section should be fully understood by all technical support staff. Controlling card technologies and protocols is extremely important for a better user experience. It also reduces or eliminates the probability of a rogue credential in the application, for example a parking garage card being read by the cafeteria system (instead of the MIFARE DESFire or Seos® card that is supported) with the data output of the reader looking like a different individual is present.

It is extremely important to understand that card polling is the process by which the reader changes RF protocols to search for each specific RF protocol and card technology. For instance, the reader will poll for cards every 100 ms (the Polling Frequency). The Polling Delay is the period that the reader will wait to turn on the next protocol and search for cards, after it has searched for cards present in the enabled protocol.

The reader will poll for cards in the following order:

1. Tech Order Table (Card Technology)
2. RF Protocol (RF Link between Card/Phone and Reader)

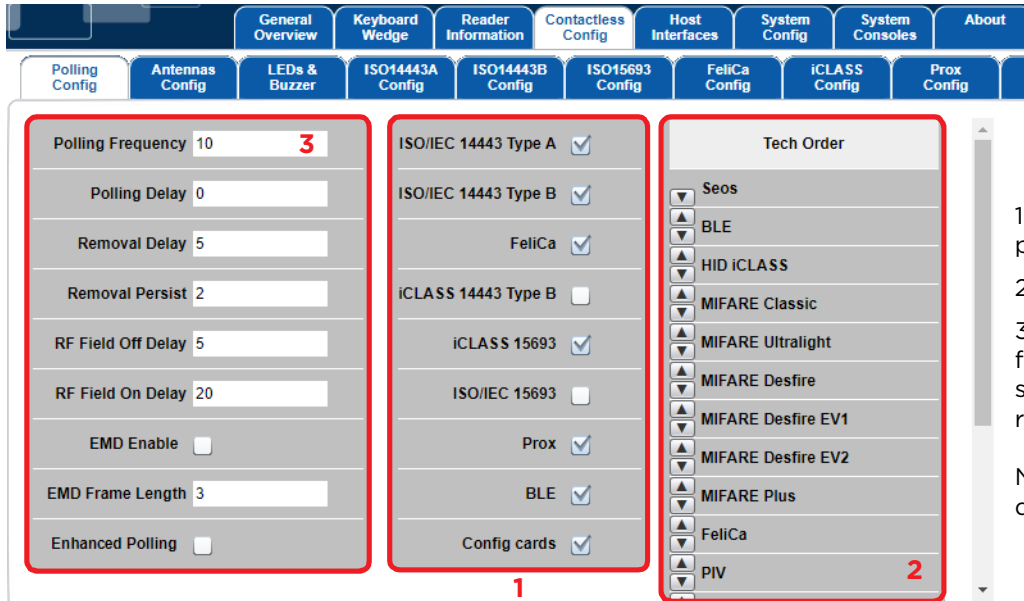
**Note:**

- The more technologies and protocols that are enabled, the slower the response of the reader.
- Use caution when disabling Config Cards within the RF Protocol section. When configuration card support is wanted at the installation site, ensure that Generic ISO14443A is also enabled without any response in the Keyboard Wedge, Card Data Selection sub-tab.

The reader polls only for the card protocols selected in the **Polling Config** tab. The reader ignores all card types cleared on this tab.

It is recommended that you create a default configuration enabling all wanted parameters, and disabling all unnecessary protocols and card technologies. This will optimize the reader response/operational timing.

**Note:** If a multi-technology card is not detected correctly, increasing the RF Field On Wait Delay (rfFieldOnWait) value may fix the problem. This value can be changed via CCID or the webserver (**Contactless Config > Polling Config > RF Field On Delay**).



1. Clear unused RF protocols.
2. Change priority table.
3. Change polling frequency to optimize speed and response of reader.

Note: HID recommends not disabling **Config cards**.

Speak to an HID Sales, Presales Engineer or Field Application Engineer for further information.

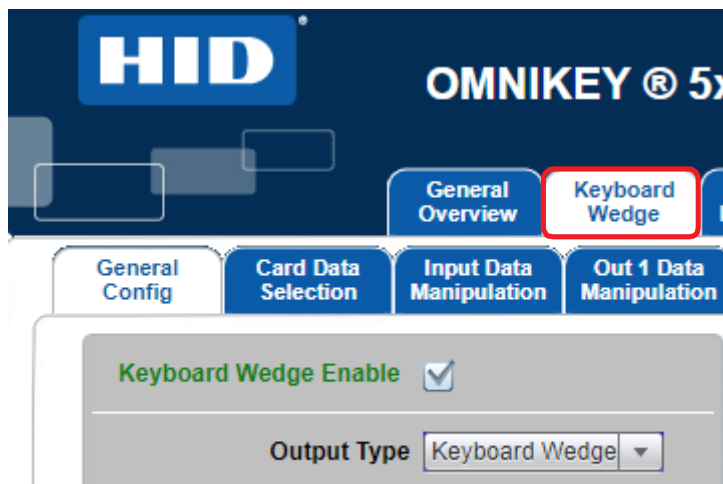
**Note:** Take account of the Polling Config settings in the **Contactless Config** menu. Disabling a card type in the **Card Type** drop-down of the **Card Data Selection** tab will not prevent the reader from polling for that card type. Deselecting the card type means only that the card data will not be processed through the keyboard wedge interface.

**Note:** It is suggested to experiment with lower RF Tx/Rx rates to better stabilize the RFID interface. Most applications will not exhibit a major difference between 106 kbps and 424 kbps. This is because the error rate becomes greater at higher Tx/Rx rates. It is recommended to change ISO14443A/B and FeliCa to lower rates.

For multi-technology cards, the card type detected is dependent on where the reader is in its polling cycle when the card is presented. Therefore, for card populations involving multi-technology cards, ensure the unwanted card type is switched off in both the **Polling Config** and **Card Data Selection** tabs.

## Keyboard wedge mode

This section describes the embedded web-based OMNIKEY® 5x27CK Reader Management tool for Keyboard Wedge. The default configuration for the OMNIKEY 5x27CK is **CCID** mode. Before using the Keyboard Wedge Mode, enable Keyboard Wedge in the **Keyboard Wedge** tab.



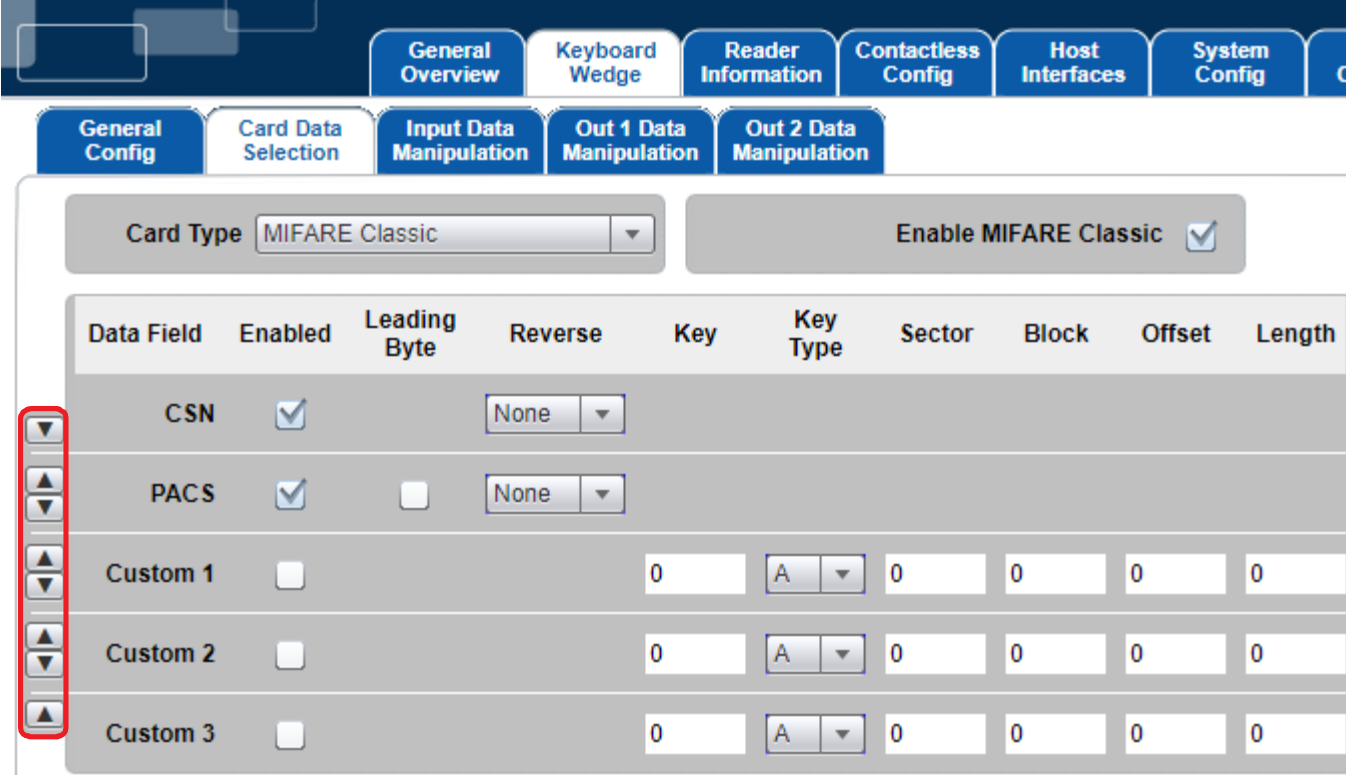
Keyboard wedge operation is a highly configurable read-only application of the reader. Care should be taken to configure the product correctly, and to enable only the card technologies and data that are needed at each installation, to lower the likelihood and/or prevent rogue credentials from being introduced to the application.

The ProcessKeyboardWedge command detailed in the *OMNIKEY 5X27CK Software Developer Guide* (5127-903) is the HID recommended implementation.

### 3.1 Card In event

The 5x27CK lets you customize your output string for a Card In event (card occurrence recognized by the reader). On the **Out 2 Data Manipulation** tab, it is possible to set **Card In Event Keystrokes** as well as **Pre-** and **Post-strokes**. The **Card Data Selection** tab allows you to select preset data to be output.

There can be multiple data fields in one output string, for example PACS bits followed by a custom data field. In this case, ensure the desired data fields are activated and fully configured. Change the order of the output string data fields by using the up/down arrow buttons (left of the data field names).



Separate data fields from each other by using pre- and post-strokes (**Out2 Data Manipulation** tab).

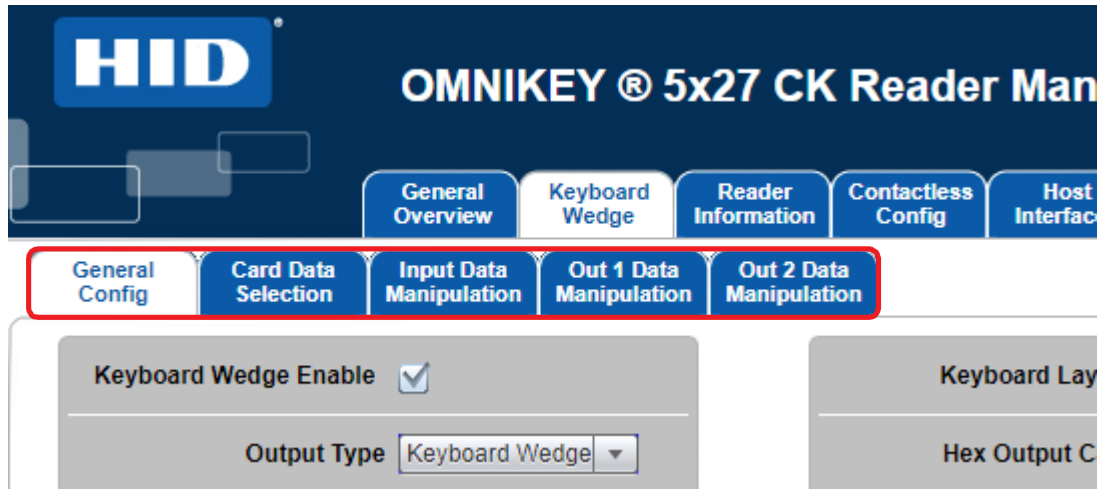
#### 3.1.1 Card Out event

The 5x27CK lets you define an output string to be sent when a card is taken from the reader. To do this, enter the desired keystrokes in **Card Out Event Keystrokes** on the **Keyboard Wedge > General Config** tab.

**Note:** This output string is sent for each card type and does not support card data.



### 3.2 Navigating the Keyboard Wedge configuration tabs

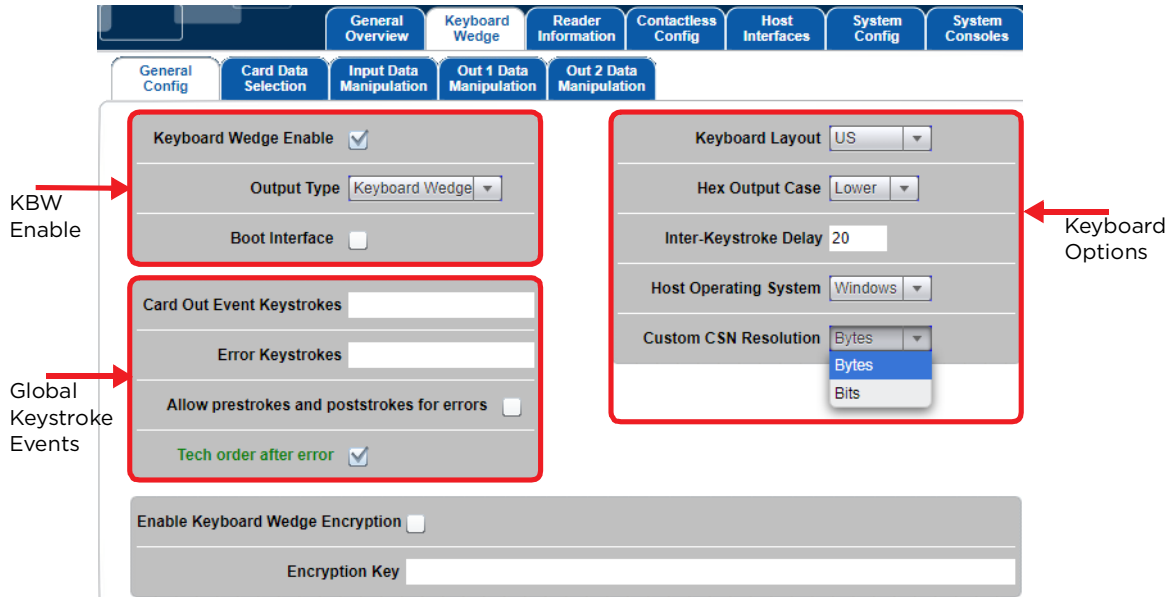


Tab	Description
<b>General Config</b>	Use this tab to enable and setup general keyboard wedge operational parameters.
<b>Card Data Selection</b>	Use this tab to enable and disable card technologies and select the data to be read from the card and reported across the keyboard interface automatically.
<b>Input Data Manipulation</b>	Use this tab to configure how the data selected in the <b>Card Data Selection</b> tab is output across the keyboard interface (Input -> input data from card to reader) (bit padding, binary/byte reverse, logic operations)
<b>Out 1 Data Manipulation</b>	Use these tabs to configure string operations performed on data received from the card, which is output across the keyboard interface. This includes the format of the output string, filtering, truncating, padding, pre- and post-strokes.
<b>Out 2 Data Manipulation</b>	

**Note:** The **Card Data Selection**, **Input Data Manipulation**, **Out 1 Data Manipulation** and **Out 2 Data Manipulation** tabs work together for the specified card technology. When changing the settings for the data output in any of the Manipulation tabs, you are changing the output configuration for the active card technology in the **Card Data Selection** tab.

### 3.3 General Config tab

The **General Config** tab allows you to configure general KBW operational settings that are not dependent on card type.



#### 3.3.1 KBW Enable options

##### Keyboard Wedge Enable

To enable the Keyboard Wedge mode, select the **Keyboard Wedge** tab and select the **Keyboard Wedge Enable** option. Return to CCID mode by clearing the **Keyboard Wedge Enable** option.

**Note:** When Keyboard Wedge is selected, the 5x27CK enumerates as a Human-Interface Keyboard device. Therefore, CCID interfaces are not available. The web interface is available in both CCID and Keyboard Wedge modes.

##### Output Type

Keyboard wedge mode includes two output types, **Keyboard Wedge** and **Custom Report**.

##### Keyboard Wedge Output

The Keyboard Wedge output is the standard. The device enumerates as a keyboard and outputs the keyboard wedge data as a series of keystrokes.

##### Custom Report Output

When Custom Report output is enabled the device enumerates as a custom HID USB device and outputs data as raw APDU as follows:

- The packet size is 40 bytes.
- 1st byte is the length of data in the packet.
- 2nd byte is the version of the report.
- The following bytes contain the keyboard wedge data.

- In cases where the data length, version, and byte length combine to less than the USB packet size (40 bytes), additional zeros are added for the remaining length.

```
000007: Bulk or Interrupt Transfer (UP), 29.01.2016 14:29:27.634 +0.180. (1. Device: USB Input Device) Status: 0x00000000
Pipe Handle: 0xe91d1a8 (Endpoint Address: 0x83)
Get 0x40 bytes from the device
01 0A 01 31 30 30 30 31 66 66 66 66 66 00 00 00 ...10001fffff...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

HID suggests the use of this mode of operation, or using the `ProcessKeyboardWedge` command in PC/SC-CCID mode when connected to a computer or other device. See *OMNIKEY 5X27CK Software Developer Guide* (5127-903).

Many people view this as a higher security option over keyboard wedge, since nobody can remove the reader and attach it to a computer to see the data being output by the reader. However, most people use only the “Card Number” from HID PACS Data for non-PACS applications throughout the enterprise (e.g. cafeteria, payment, library, secure print, etc.) so security is dependent upon the application and risk model.

### Boot Interface

The **Boot Interface** option allows the device to advertise support for the keyboard boot interface in its HID device descriptor when it enumerates as a keyboard device. If enabled, the device is operational on host systems that only have minimal USB device handling, without support for full USB descriptor parsing.

## 3.3.2 Global keystroke events

These keystroke events are not card type dependent.

### Card Out Event Keystrokes

The OMNIKEY 5x27 reports the keyboard strokes as configured when a supported card is presented and removed from the reader. These events are referred to as Card Out (removed) events.

Card Out defines a set of keystrokes that are sent over the keyboard interface when a card is removed from the reader. Due to the card removal from the reader, those keystrokes are generic (card-independent) and apply to all card types supported by the reader. If the text box is left blank, no action is performed by the OMNIKEY 5x27 reader when a card is removed from the field.

### Error Keystrokes

The OMNIKEY 5x27 reports the configured error keystrokes when the reader fails to access, buffer, process and report a specific data field as configured in the **Card Data Selection** tab. Possible instances of a failure might be:

- Multiple RFID tokens of the same ISO protocol are presented simultaneously to the reader
- The card that is selected does not contain the data wanted.
- The key loaded and or selected in the reader does not match the key loaded onto the RFID token and access to the data field is denied.

### Allow prestrokes and poststrokes for errors

When enabled, the pre-strokes and post-strokes configured in the **Card Data Selection** tab will be output by the reader upon an error occurring.

### Tech order after error

If this option is enabled, when a card data processing error occurs the OMNIKEY 5x27 reader will continue processing the card type in the order defined in **Contactless Config > Tech Order** tab.

The intended use of this setting is for installations that use a mix of technology cards within the enterprise.

**Note:** When enabled, the output is delayed until all the card data is processed. If a failure occurs, no data is output from the reader (for the card type on which the error occurs) including pre- and post-strokes, as if no card were presented. This prevents the host system from having to process the data unnecessarily. Note also that this may lead to a flickering ATR display if all the card data cannot be correctly processed.

## 3.3.3 Keyboard options

### Keyboard Layout

This selection compensates for differences in regional keyboard layouts (for example, different interpretation of Y key on a US and DE keyboard). This setting must be adjusted to the actual setting of the host system in which the 5x27CK is connected. The following layouts are built into the reader:

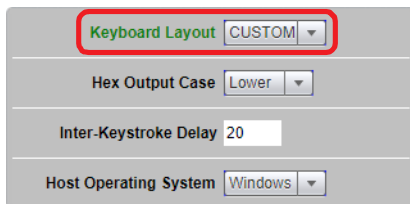
- France
- Germany
- United Kingdom
- United States
- Custom

Example: A **Y** in the keyboard wedge layout **US** generates a **Z** on a host-PC using the German keyboard layout. Only when the keyboard wedge is configured to **DE** will the **Y** be interpreted correctly as a **Y** on the host-PC.

### Custom Layout

The reader allows for any keyboard layout to be used with the reader. To use such a layout, follow these steps:

1. Create a keyboard layout file using Microsoft Keyboard Layout Creator.
2. Send the created file to HID tech support. They will convert this file to an encrypted file in the correct format for the reader to interpret.
3. Open the **OK5x27CK** webserver and navigate to the **Keyboard Wedge** tab.
4. Select the **CUSTOM** option from the **Keyboard Layout** drop-down menu.



5. Navigate to the **System Config** tab.
6. Click **Apply Changes**.

7. For the **Load Key Layout** setting, click **Browse** and select the layout file provided by technical support.
8. Click **Load Key Layout**.



### Hex Output Case

This option specifies whether hexadecimal output is lower or upper case. The setting applies to all card types.

### Inter-Keystroke Delay

This setting allows you to set a delay (in milliseconds) between consecutive keyboard strokes.

### Host Operating System

This setting allows the Keyboard Wedge to correctly interpret extended ASCII characters, depending on the host operating system. Each supported operating system has a different input method for extended characters.

### Custom CSN Resolution

This option specifies the resolution of the **CSN Custom  $n$**  field in the **Card Data Selection** tab (see *Section 3.4: Card Data Selection tab*).

- Bytes: Custom CSN manipulation at byte level (default):

Data Field	Enabled	Reverse	Offset (Bytes)	Length (Bytes)
CSN Custom 1	<input type="checkbox"/>	None	0	0
CSN Custom 2	<input type="checkbox"/>	None	0	0

- Bits: Custom CSN manipulation at bit level:

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
CSN Custom 1	<input type="checkbox"/>	None	0	0
CSN Custom 2	<input type="checkbox"/>	None	0	0

## 3.3.4 Keyboard wedge encryption

### Enable Keyboard Wedge Encryption

This enables the keyboard wedge output encryption, which uses a 128-bit AES algorithm with SIV (<https://tools.ietf.org/html/rfc5297>). That is why the Keyboard Wedge encryption key has 32 bytes, as opposed to standard AES128 16-byte key. There is a default key in the reader that can be replaced by a user key as described below. In normal keyboard wedge mode over USB, the output is encoded as hex characters. For all other modes the output is binary.

Sample code written in C is available demonstrating decryption of the output using the default key.

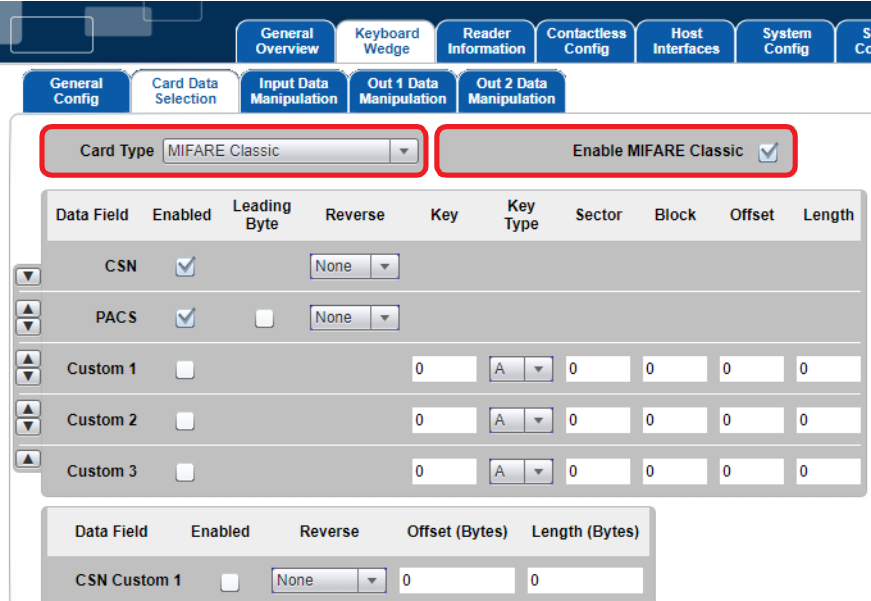
### Encryption Key

Enter a 32 byte encryption key in hex to replace the default key in the reader. This is write only and there is no way to read the key back from the reader.

### 3.4 Card Data Selection tab

The **Card Data Selection** tab allows you to set the keyboard wedge actions once a card is detected by the reader. Card In events are customizable depending on the detected card type. See *Section 1.4: Supported RFID technologies*). To configure Keyboard Wedge output for specific card type:

1. Select **Card Type** from the drop-down menu.  
All supported cards are available for configuration in the **Card Type** drop-down menu on the **Card Data Selection** tab. The default configuration has all card types active (except Generic ISO14443A) and preset data fields are sent upon card detection.
2. Enable and Disable Card Type Processing.  
Deselect cards through the web server by clearing the **Enable** check box on each card page.



### 3.4.1 Configure data fields for each card type.

#### Preset Data Fields

Preset data fields represent the card's pre-configured data objects. For the 5x27CK, those are the PACS-Bits and CSN. Memory areas and key configuration is preset in the 5x27CK. Therefore, no configuration is required to access those data fields.

Field	Description
CSN	The Card Serial Number (CSN) is a data string which identifies a Smart card chip.
PACS	The PACS Data is used in Physical Access Control Systems as the credential to identify an individual within a controlled card population. This field is intended to be used when the system is designed to be format agnostic or when the system handles format data such as in a PACS application.
Custom n	Custom data fields are used to access any piece of data programmed on a card outside the CSN and PACS Data.
PACS Custom	<p>PACS Custom allows you to parse the PACS Data into multiple data fields. The most common data fields are:</p> <ul style="list-style-type: none"> <li>■ Facility Code</li> <li>■ Card Number</li> <li>■ Site Code</li> <li>■ City Code</li> <li>■ OEM Code</li> </ul> <p>The PACS Format Fields used are dependent upon the PACS Data Format.</p>

**Note:** CSN is not available for Prox cards.

**Note:** When using PACS Custom, HID suggests using more than one PACS format field. The OMNIKEY 5x27 readers have been updated to support up to 4 fields to support parsing 2 fields of 2 different formats (firmware version 04000000 and higher).

#### Card Serial Number (CSN)

The CSN is open and in the clear. This means that the CSN is not secure and is open to copy and replay. With new NFC mobile devices, it is possible for the CSN to be copied and replayed with relative ease. To better meet security threats such as NFC enabled mobile devices, Next Generation Smartcards and NFC mobile devices use a Random Card Number in place of the CSN. When the card type or card emulation uses a Random Card Number, it will be output by the reader. Thus, for these technologies, CSN is not an adequate credential to be used for any application. For instance, the Seos® CSN will output a random 4 byte number.

HID suggests migrating away from the CSN as the credential whenever possible.

#### Other Considerations for CSN

When leveraging a CSN credential based PACS database, the application must often support CSN data manipulation to match the database. The OMNIKEY 5x27 always provides the complete CSN transferred during the anti-collision and card selection process, in accordance with Smartcards ISO standards.

#### PACS

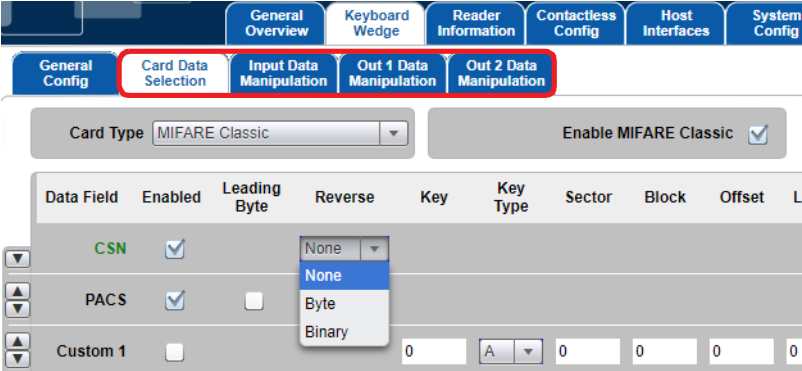
The PACS data field is often used to create a PACS format agnostic system or in cases when an entity does not wish to disclose their PACS format.

### 3.5 Data manipulation

OMNIKEY 5x27CK in Keyboard Wedge mode enables reading raw data from the card, and various modifications of the output string.

**Note:** Before Gen1 SP3, all pre- and post-stroke, card in, card out and error fields were limited to 7 characters (normal and special combined). From SP3 onwards, each one can be up to 250 characters. However, the total memory used by these characters must not exceed 1024 bytes, and there is a formatting overhead of 5 bytes per item. Empty entries do not incur any overhead. For example, eight 123 character strings would exactly fill all of the memory available.

**Note:** The general rule for data manipulation is that the data modifications are processed left to right, according to the tabs within the **Keyboard Wedge** tab, and from left to right within each tab. This means that the modifications from the **Card Data Selection** tab are applied first (Leading Byte > Reverse) followed by the modifications defined in the **Input Data Manipulation** tab, etc.





## 3.6 Input Data Manipulation tab

### 3.6.1 PACS Leading Byte

PACS data is a binary structure and therefore, normally not a full byte-length-value (8 bits = 1 byte). For example, the H10301 26-bit Wiegand PACS format must be padded to 32 bits before the binary to HEX conversion can take place.

The normal HEX data is simply left padded to the nearest full-byte-length with binary 0s. When PACS Leading Byte is enabled, the binary PACS data is right padded with binary 0s and the number of padding bits is encoded as the PACS Leading Byte.

Example (H10301 26-bit Wiegand PACS Format):

Data on Card	01 10010000 00100111 00010010
HEX Output	01 90 27 12
HEX Output with PACS Leading Byte Enabled	06 64 09 C4 80
Breaking HEX string into binary PACS Data Output = <b>066409C480</b> Binary = <b>00000110 01100100000010011101010010 000000</b> <b>Number of bits that are right padded onto the binary PACS data</b> <b>Raw data from card</b> <b>Padded zeros</b>	

Data on Card	1111111111111111100000000000000000000010
HEX Output	07FFE00002
HEX Output with PACS Leading Byte Enabled	05FFFC000040
Breaking HEX string into binary PACS Data Output = <b>05FFFC000040</b> Binary = <b>00000101 111111111111101000000000000010 000000</b> <b>Number of bits that are right padded onto the binary PACS data</b> <b>Raw data from card</b> <b>Padded zeros</b>	

**Note:** PACS Leading Byte was added to the OMNIKEY 5x27 to support the HEX data output, only to enable the OEM application to easily determine the actual PACS data programmed on the card.

**Note:** The PACS Leading byte will affect all data output formats.

### 3.6.2 Binary Reverse

Bits of raw binary data from card are reversed.

PACS data:

10 0000 0010 0000 0000 1100 1110 (0x020200CE)

PACS data with Binary Reverse:

0111 0011 0000 0000 0100 0000 01 (0x01CC0101)

### 3.6.3 Byte Reverse

Byte Reverse reverses the standard read order of the card data. The order is changed on raw byte-level data as shown below.

PACS data:

0000 0110 0110 0100 0000 1001 1100 0100 1000 0000 (0x066409C480)

PACS data with Byte Reverse:

1000 0000 1100 0100 0000 1001 0110 0100 0000 0110 (0x80C4096406)

The reverse order supports all output formats (BIN, HEX, DEC, BCD and ASCII). Although, HEX output with the PACS Leading Byte enabled is when it is mostly used.

#### Example (H10301 26-bit Wiegand PACS Format):

Output Format	H10301 Output
BIN (Reverse Disabled)	00000110011001100000010011100010010000000
BIN (Reverse Enabled)	1000 0000 1100 0100 0000 1001 0110 0100 0000 0110
HEX (Reverse Disabled)	066409C480
HEX (Reverse Enabled)	80C4096406
DEC (Reverse Disabled)	27448165504
DEC (Reverse Enabled)	553044763654 $0x80 \cdot (2^{32}) + 0xC4 \cdot (2^{24}) + 0x09 \cdot (2^{16}) + 0x64 \cdot (2^8) + 0x06$
BCD (Reverse Disabled)	00100111010001001000000101100101010100000100
BCD (Reverse Enabled)	0101 0101 0011 0000 0100 0100 0111 0110 0011 0110 0101 0100 5 5 3 0 4 4 7 6 3 6 5 4

Note that older firmware versions had slightly different behavior:

- Firmware 02000000: Only Byte Reverse is possible and it applies only to custom data fields. PACS and CSN bits will not be affected by this command.
- From firmware version 03000000 and greater, reverse applies to all data fields (only Byte Reverse).

### 3.6.4 Bit Padding

The Bit Padding feature allows you to add specified leading or trailing bits to the raw data received from card. This modification makes it possible to get data with constant, specified length. Data modified in this way might be useful when performing any logic operations.

Additional bits might be added before or after the raw data. The length parameter specifies the number of bits to be padded.

PACS Custom (offset: 7, length: 14):

01000000000110 (0x1006)

Value: 1, Direction: Leading, Length: 2:

**11**01000000000110 (0xD006)

Value: 0, Direction: Trailing, Length: 2:

010000000001**1000** (0x4018)

**Note:** This is a new feature added from firmware version Gen 2, SP2. Before this, only String Padding functionality was available.

### 3.6.5 Logic Operations

Logic operations allow you to perform one of three logical operations on data received from the card:

- AND
- OR
- XOR

This feature was introduced in OMNIKEY 5x27CK to allow bit inversion on raw card data (XOR).

The bit mask is in HEX format and output data is always the same size as the input data. If the mask is shorter than the input data, the mask is implicitly padded with leading zero bits to match the input data size. If no mask is set (mask = "") the logical operation will not be performed.

Data received from card, PACS Custom (offset 7, length: 14)

01000000000110 (0x1006)

Operation: **AND**, Mask: 0x93C2 (1001001111000010):

01000000000010 (0x1002)

*Note: Mask is bigger than data, but output matches input data size*

Operation: **OR**, Mask: 0x93 (10010011 -> padded to: 00000010010011)

01000010010111 (0x1097)

*Note: Mask is smaller and is padded with leading zero bits to match input data size.*

Operation: **XOR**, Mask: 0xFFFF (1111111111111111)

10111111111001 (0x2FF9)

## 3.6.6 Out 1 Data Manipulation tab

### 3.6.6.1 String Format

This option allows you to define the format of the output data. There are five possible output formats:

- BIN
- HEX
- DEC
- BCD
- ASCII

The same data presented in different formats:

HEX: 484944476c6f62616c

ASCII: HIDGlobal

BCD:0101001000001000011101101001010100010111010110000011001010010101000001110011

BIN: 010010000100100101000100010001110110110001101111011000100110000101101100

DEC: 5208769517583295073

### 3.6.6.2 String Filtering

The String Filtering function allows specified characters to be removed from raw data. Any character supported by the Keyboard Wedge can be removed.

Card data output:

HIDGlobal

String Filtering, Char: "D", Direction: Leading:

HIGlobal

Previous versions of firmware had slightly different behavior:

- Firmware up to 03000000:  
Filter a byte (entered as decimal ASCII code) from raw data.
- Firmware 03000000 or higher:  
Direction: Leading = filter bytes from the start of raw data, Trailing = filter bytes from end of output data.
- Firmware 04000000 or higher:  
The filter character no longer needs to be entered as a decimal coded ASCII value and is entered by the desired keyboard character.

### 3.6.6.3 String Truncating

String Truncating is a feature present in the OMNIKEY 5x27CK reader since Gen2 SP2 (firmware version 01.02.00f7). It allows part of the output string to be truncated (cut). The current implementation makes it possible to:

- Cut off: Return the data/characters defined by Offset and Length.
- Remove: Delete the data/characters defined by Offset and Length and return the remainder.

**Note:** Indexing starts with 0. The first character in the string has index 0.

Input data: PACS Custom (offset: 7, length 14):

01000000000110 (0x1006)

Operation: Remove, Offset: 5, Length: 4, String Format: BIN:

IN: 01000**0000**00110

OUT: 0100000010

Operation: Remove, Offset: 1, Length: 2, String Format: DEC:

IN: 4**10**2

OUT: 42

Operation: Cut off, Offset: 1, Length: 2, String Format: HEX

IN: **100**6

OUT: 00

### 3.6.7 Out 2 Data Manipulation tab

#### 3.6.7.1 String Padding

String Padding (called *Padding* before Gen 2 SP2, firmware version 01.02.00f7) allows any specified character to be added to the input string. Additional characters can be pasted at the beginning or end of the data.

Input data: PACS Custom (offset: 7, length 14):

010000000000110 (0x1006)

String Padding, Char: f, Direction: Leading, Length: 2, String Format: HEX:

1006ff

Padding behavior for previous firmware versions:

- Firmware 0300000:

Padding bytes are added to the raw data.

Byte: ASCII character value (in decimal) to add to output string. It is output depending on the Format as specified above. So 48 would be output as 30 in hex or 0 in decimal. Binary is a special case, where only 0, 1, 48 or 49) are allowed - other values will be displayed as 1.

Direction: Leading = add padding to start of string, Trailing = add padding to end of string.

Length: Number of output characters to pad out to. This is format-independent, so entering 10 gives you 10 hex digits, 10 decimal digits, 10 ASCII characters, 10 binary bits, etc.

- Firmware 04000000 or higher:

This feature is changed to support fixed data output requirements. Given this, if the number of padded characters is equal or less than the output string, the padded characters will not be added.

In addition, you may now place the actual character in the Char text box instead of its ASCII equivalent.

**Note:** This should be the last setting configured in the **Out 2 Data Manipulation** tab.

#### 3.6.7.2 Pre- and Post-strokes

For each data field, it is possible to define a set of characters to be output before and after the data. All supported ASCII characters, as well as special control characters (see *Section 3.7: Supported keystroke & command characters*) can be used as keystrokes.

Custom Data, String Format: ASCII, Pre-strokes: "CardData: ", Post-strokes: "[ENTER]OMNIKEY"

Input:

HIDGlobal

Output:

CardData: HIDGlobal

OMNIKEY

## 3.7 Supported keystroke & command characters

### 3.7.1 Supported printable characters

All normal printable keyboard ASCII characters are supported by the OMNIKEY 5x27.

### 3.7.2 Pre- and post-stroke supported control characters

In most cases, keyboard stroke data (Pre and Post, or both) are strings of standard ASCII characters. In addition, use control characters, such as the Enter key. Enclose the control character (key) in brackets [ ], for example, [ENTER].

#### IMPORTANT:

- For confirming post- or pre-keystrokes in firmware versions below 02000000, press Enter, for the reader to perform validity check on the keystrokes.
- For firmware versions 02000000 or above, pressing Enter **is not required**, the reader performs a validity check automatically once the focus is taken from the data field (for example, by pressing the Tab key or clicking another data field).
- For valid keystrokes, the font color turns from black to green. The text color remains green until you click **Apply Changes** in the **System Config** tab.
- In case the validity check fails, the font color turns red.

Possible failures include the following:

- Incorrect syntax in control commands.
- Exceeding the max length per data field, which is 250 characters.

The following table lists all supported control characters.

**Note:** Control characters must be capital letters.

KBW allows you to combine keystrokes with ASCII characters to allow shortcuts on the computer. For example, [ALT]F[CTRL]N[ENTER] creates a new text file when the Notepad application is active on the computer.

**Supported Control Characters**

<b>Control Character / Key</b>	<b>Abbreviation</b>
End	END
Enter	ENTER
Esc	ESC
Cursor down	DOWN
Cursor up	UP
Cursor left	LEFT
Cursor right	RIGHT
Space	SPACE
Tab	TAB
F1	F1
...	...
F12	F12
Shift	SHIFT
Ctrl	CTRL
Alt	ALT
Delete	DEL
Windows	GUI



In normal keyboard wedge mode over USB, [CTRL] preceding another character will generate the corresponding keypress. However, when using the serial UART output, the corresponding ASCII control character will be sent according to the following table.

Pre / Post Stroke Characters	ASCII Control Character	Dec	Hex	Pre / Post Stroke Characters	ASCII Control Character	Dec	Hex
[CTRL]@	NUL	0	00	[CTRL]P	DLE	16	10
[CTRL]A	SOH	1	01	[CTRL]Q	DC1	17	11
[CTRL]B	STX	2	02	[CTRL]R	DC2	18	12
[CTRL]C	ETX	3	03	[CTRL]S	DC3	19+	13
[CTRL]D	EOT	4	04	[CTRL]T	DC4	20	14
[CTRL]E	ENQ	5	05	[CTRL]U	NAK	21	15
[CTRL]F	ACK	6	06	[CTRL]V	SYN	22	16
[CTRL]G	BEL	7	07	[CTRL]W	ETB	23	17
[CTRL]H	BS	8	08	[CTRL]X	CAN	24	18
[CTRL]I	TAB	9	09	[CTRL]Y	EM	25	19
[CTRL]J	LF	10	0A	[CTRL]Z	SUB	26	1A
[CTRL]K	VT	11	0B	[CTRL][	ESC	27	1B
[CTRL]L	FF	12	0C	[CTRL]\	FS	28	1C
[CTRL]M	CR	13	0D	[CTRL]]	GS	29	1D
[CTRL]N	SO	14	0E	[CTRL]^	RS	30	1E
[CTRL]O	SI	15	0F	[CTRL]_	US	31	1F

For example, to generate a CR and LF in the serial output, place [CTRL]M[CTRL]J in the pre or post stroke string.

### 3.7.3 Extended ASCII Character Set (OK5427 Gen2/OK5127CK Mini/OK5127CK Reader Core onwards)

From OK5427 Gen 2 / OK5127 Mini SP2 / OK5127CK Reader Core onwards, the above pre and post stroke strings support an extended ASCII character set, which is listed in a table in *Appendix B*.

Characters entered into the web server text boxes are converted to their equivalent ASCII code.

- Those characters that have a standard ASCII code from 32 to 127 are stored as such.
- Any characters that are in the extended table in the appendix are converted to the equivalent ASCII code in the table.
- Any other characters are ignored.

When an extended character is output via the normal keyboard wedge over USB, the character will be “typed” according to the host operating system selected on the **General Config** tab.

Host Operating System	Method
Windows	The [Alt] key will be held whilst the ASCII code is typed on the number keypad.
Linux	[Shift][Ctrl] U is pressed followed by the Unicode value in hex.
MacOS	The keyboard must be in Unicode Hex Input mode. The [Alt] key is held whilst the Unicode value is typed in hex.

### 3.7.4 Reader command keystrokes (controlling reader behavior)

#### 3.7.4.1 [PAUSE xxx]

The PAUSE character places the OMNIKEY reader into a hold state where it will not process any cards. This is to allow the host system to process the card data received by the reader and perform additional functions before possibly receiving another dataset from the reader.

The value setting is 1 = 100 milliseconds (decimal) as follows. Note that the following example shows a pause of 2 seconds.

**Note:** The following example causes these events to occur before another card can be processed:

- Outputs the CSN data followed by [ENTER] and performs the LED/buzzer sequence.
- Delays for a 2 second wait period.
- Outputs Custom 1 data

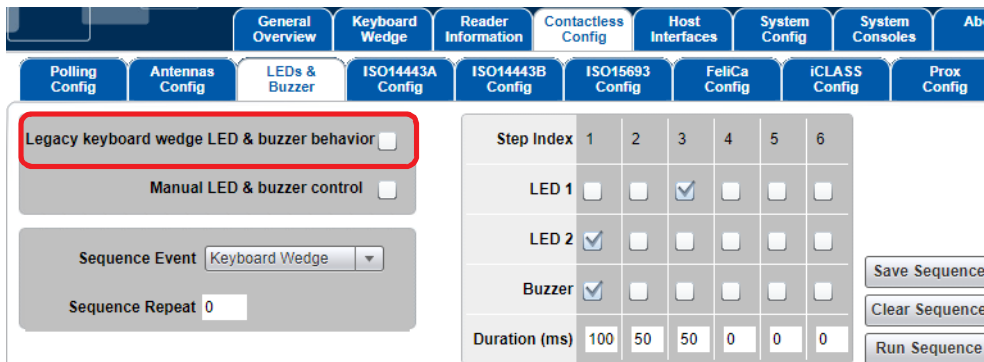
The screenshot shows the HID web interface configuration for Card In Event Keystrokes. The 'Card In Event Keystrokes' field is set to [LED\_BUZZ]. The 'Datafield' table shows the following configuration:

Datafield	String Padding	Prestrokes	Poststrokes
CSN	Char 0, Direction: Leading, Offset: 0	CSN:	[ENTER][PAUSE 20]
PACS	Char 0, Direction: Trailing, Offset: 0		
Custom 1	Char 0, Direction: Leading, Offset: 0	Data:	[ENTER]

### 3.7.4.2 [LED\_BUZZ]

The LED\_BUZZ character provides the capability to control the LED and Buzzer sequence timing to provide a customized user experience. Each instance of an LED\_BUZZ character is placed in the pre or post strokes field. The Card Access LED and Buzzer sequence will initiate as configured in the **LEDs & Buzzer** tab in the **Contactless Config** tab.

To enable this feature, the **Legacy keyboard wedge LED & Buzzer behavior** option must be cleared.



## 3.8 Secure messaging on MIFARE DESFire EV2

Support for MIFARE DESFire EV2 cards is provided from SP2. If you need to use secure messaging with these cards, you must set the **Card Data Selection** as follows:

- **Auth** check box selected (Auth enabled).
- **File Comms** set to **MACed** or **Encrypt**.
- Key type set to **AES** (Encryption option).

## 3.9 Keyboard Wedge output via UART

The OMNIKEY 5x27-Mini supports a UART interface. See *5127-903 - OMNIKEY 5x27CK Software Developer Guide* (5127-903) for details about setting up this interface and communication.

The serial protocol features keystrokes output.

The HEX output across the UART mirrors the ASCII characters output across the keyboard emulation. The only difference is the UART output also shows non-printable ASCII characters, to include all special characters such as [LED\_BUZZ], [ENTER], [TAB], etc.

Example UART output:

```
4F 4D 4E 49 4B 45 59 20 35 78 32 37 20 43 4B 20 4D 69 6E 69 20 3D 20 45 6D 75 6C 61 74 69 6E 67
20 61 20 6B 65 79 62 6F 61 72 64 0A 69 43 4C 41 53 53 20 43 53 4E 3A 32 62 63 32 34 35 30 31 66 38
66 66 31 32 65 30 0A 69 43 4C 41 53 53 20 48 49 44 20 50 41 43 53 20 44 61 74 61 3A 31 30 30 30 31
66 66 66 66 66 0A
```

- OMNIKEY 5x27 CK Mini = Emulating a keyboard
- iCLASS® CSN:2bc24501f8ff12e0
- iCLASS HID PACS Data:10001ffff
- [ENTER]

### 3.10 Maximum output size

The maximum output size of custom data in HEX format is limited to 255 bytes in total.

# Chapter 4

## Custom Report mode

Custom Report mode requires that KBW is enabled within the reader. It outputs the configured data as raw HEX (ASCII) over the basic USB Human Interface Device class interface.

Custom Report is considered to be a higher security interface option, as somebody must have additional knowledge to interact with a HID Class device.

Custom Report is not mirrored across the UART.

Custom Report is a simplistic interface that does not completely mirror Keyboard Wedge. Specifically, the output exposes the operation of the reader. Card In Event Keystrokes, Prestrokes, Data, Poststrokes and Card Out Events are all separate executions by the reader.

### 4.1 Example Custom Report output across USB HID interface

Custom Report output requires knowledge of ASCII and reader configuration at a minimum. Developers must also understand the USB HID Specification. In the following example, the reader is configured to provide the following output:

- Card In Event = Card In[TAB][ENTER]
- Prestroke Event = iCLASS® HID PACS Data:[TAB]
- Data = PACS Data output in HEX with the PACS Leading Byte
- Poststroke Event = [TAB][ENTER]
- Card out Event = Card Out[ENTER][ENTER][TAB]

The screenshot shows a configuration window with five tabs: General Config, Card Data Selection, Input Data Manipulation, Out 1 Data Manipulation, and Out 2 Data Manipulation. The 'General Config' tab is active. It contains the following settings:

- Keyboard Wedge Enable:** Checked (checkbox).
- Output Type:** A dropdown menu set to 'Custom Report'.
- Boot Interface:** Unchecked (checkbox).
- Card Out Event Keystrokes:** A text input field containing 'Card Out[ENTER][ENTER]'. The text is highlighted in green.
- Error Keystrokes:** An empty text input field.

General Config Card Data Selection Input Data Manipulation Out 1 Data Manipulation Out 2 Data Manipulation

Card Type: HID iCLASS Enable HID iCLASS

Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Book	Page	Block	Offset	Length
CSN	<input type="checkbox"/>		None							
PACS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	None							

General Config Card Data Selection Input Data Manipulation Out 1 Data Manipulation Out 2 Data Manipulation

Card Type: HID iCLASS Hex Output Case: Lower

Datafield	String Format	String Filtering		String Truncating		
CSN	HEX	<input type="checkbox"/> Char 0	Direction: <span>Leading</span>	<input type="checkbox"/> Type: <span>Cut off</span>	Offset: <span>0</span>	Length: <span>0</span>
PACS	HEX	<input type="checkbox"/> Char 0	Direction: <span>Leading</span>	<input type="checkbox"/> Type: <span>Cut off</span>	Offset: <span>0</span>	Length: <span>0</span>

General Config Card Data Selection Input Data Manipulation Out 1 Data Manipulation Out 2 Data Manipulation

Card Type: HID iCLASS Card In Event Keystrokes: Card In[TAB][ENTER]

Datafield	String Padding			Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char <span>0</span>	Direction: <span>Leading</span>	Offset: <span>0</span>		
PACS	<input type="checkbox"/> Char <span>0</span>	Direction: <span>Leading</span>	Offset: <span>0</span>	<span>iCLASS HID PACS Data:[TAB]</span>	<span>[TAB][ENTER]</span>

### 4.1.1 Output example:

#### Packet 1 = Card In Event (Text + TAB + ENTER)

```
01 09 01 43 61 72 64 20 49 6E 09 02 00 00 00 00 ...Card In.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

#### Packet 3 = Prestroke Event (Text + TAB)

```
01 16 01 69 43 4C 41 53 53 20 48 49 44 20 50 41 ...iCLASS HID PA
43 53 20 44 61 74 61 3A 09 00 00 00 00 00 00 00 CS Data:.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

#### Packet 5 = Data Output Event (PACS Data Output)

```
01 0C 01 30 33 38 30 30 30 66 66 66 66 66 38 00 ...038000fffff8.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

#### Packet 7 = Poststroke Event (TAB + ENTER)

```
01 02 01 09 02 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

#### Packet 9 = Card Out Event (Text + ENTER + ENTER + TAB)

```
01 0B 01 43 61 72 64 20 4F 75 74 02 02 09 00 00 ...Card Out ....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

#### Notes:

- Wait approximately 10 seconds before removing the card. The packet timing between packet 7 and 9 (Poststroke and Card Out Events) is dependent on this timing.
- Notice that the even packets are not shown; these packets are the host issuing the HID Get Report Command.
- In Custom Report mode, the [ENTER] special character is ASCII “Start of Text”, while in Keyboard Wedge mode/UART, the ASCII representation for the special character is 0x0A, which is “NL line feed, new line”. This highlights to software developers that differences exist.
- Card In, Prestroked, Data output, Poststroked and Card Out events are all treated as separate interrupt transfers.
- The reader firmware is designed to read, process, and buffer the data, then output Prestroked (Event 1), Data (Event 2), and Poststrokes (Event 3). Notice that this event execution is separate from Card In and Out Events, which are directly tied to inserting and removing the card from the reader’s magnetic field.

This page intentionally left blank.



# Chapter 5

## Additional settings

### 5.1 LEDs & buzzer

This section covers how to configure the LED and Buzzer action settings for card events during a card access event.

#### 5.1.1 Navigating the LEDs & Buzzer tab

The following illustration refers to steps in *Section 5.1.3: Configuring the LED and buzzer behavior*:

The screenshot shows the HID OMNIKEY 5x27 CK Reader Management interface. The top navigation bar includes tabs for General Overview, Keyboard Wedge, Reader Information, Contactless Config, Host Interfaces, System Config, System Consoles, and About. Below this is a sub-menu with tabs for Polling Config, Antennas Config, LEDs & Buzzer, ISO14443A Config, ISO14443B Config, ISO15693 Config, FeliCa Config, iCLASS Config, Prox Config, and BLE Config. The LEDs & Buzzer tab is active, showing a configuration table for LED and Buzzer behavior across six steps. The table is highlighted with a red border. Red arrows point to specific elements: Step 1 (Sequence Event), Step 2 (LED 1), Step 3 (Run Sequence), Step 4 (Save Sequence), Step 5 (Sequence Repeat), and Step 6 (Manual LED & buzzer control).

Step Index	1	2	3	4	5	6
LED 1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LED 2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Buzzer	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Duration (ms)	100	50	50	0	0	0

Buttons: Save Sequence, Clear Sequence, Run Sequence

#### 5.1.2 Legacy keyboard wedge LED & buzzer behavior

The legacy LED and buzzer operation executes the Card Access Step Index configuration settings at the beginning, and another shortly following the first. The legacy LED and buzzer behavior is disabled by default, as some users found this to be confusing.

**Note:** Make sure that the legacy LED and buzzer behavior is disabled to support the [LED\_BUZZ] command character.

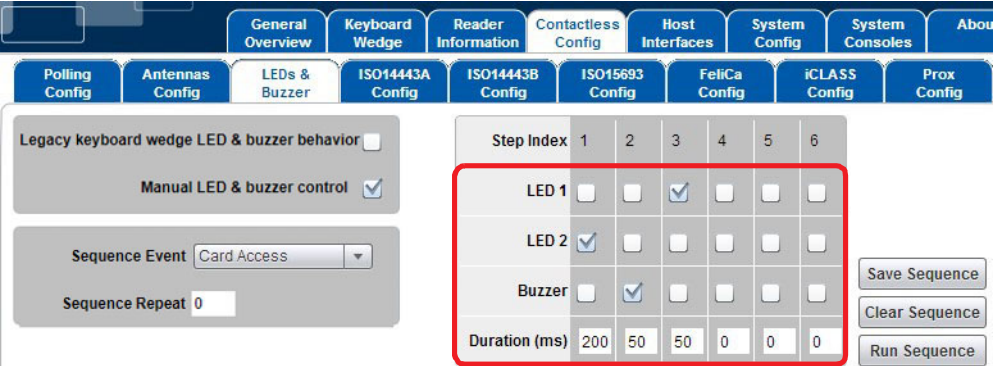
### 5.1.3 Configuring the LED and buzzer behavior

1. Select an event from the **Sequence Event** drop-down list. The following descriptions clarify each sequence event.

Sequence Events	Description
USB Ready	The LED and buzzer sequence that occurs if the OMNIKEY® 5x27 successfully enumerates with the OS and is ready.
Card Access	The LED and buzzer sequence that is initiated via the legacy LED and Buzzer behavior (when enabled).
No USB	The LED and buzzer sequence that occurs if the OMNIKEY 5x27 fails to enumerate with the OS.
Keyboard Wedge	This is the LED sequence that is triggered when the keyboard wedge encounters the special [LED_BUZZ] character in a pre-stroke, post-stroke, card in-strokes, card out-strokes or error strokes field.

2. Configure the LED and buzzer sequence and timing:
  - Sequence: Select a check box for each sequence parameter: LED 1, LED 2, and Buzzer.
  - Timing: Enter the duration in milliseconds, for each Step Index in the **Duration (ms)** field, to define the duration for the event.

Example: Upon Card Access, start with LED color 2 for 200 ms, then sound the buzzer for 50 ms, followed by LED 1 for 50 ms.



**Note:** Always ensure that you end the card access sequence with the beginning state of the USB Ready Sequence, to ensure a smooth transaction, and that the colors are reset to the USB Ready state as shown above. This will prevent an unwanted buzzer/LED state remaining after the sequence has completed. See *Section 5.1.3.1: Incorrect LED/buzzer sequence*.

3. Select **Run Sequence** to test the sequence.
 

Observe the LED and buzzer behavior to make sure everything is set up correctly. Repeat step 2 and this step as needed.
4. Once the sequence and timing is correct, select **Save Sequence** to save the sequence to memory.
5. In the **Sequence Repeat** field, enter the number of times, from 0 to 255, that the LED and buzzer sequence will repeat.
 

**Note:** 255 means that this is a permanent change. Thus the value of 255 should only be used for static events such as USB Ready and No USB.
6. Clear **Manual LED & buzzer control** to allow the sequence to run automatically on every event.

### 5.1.3.1 Incorrect LED/buzzer sequence

In the following example, clicking **Run Sequence** will cause a constant buzzing, because the Card Access sequence ends with a buzzer activation. It must end with the same LED 1, LED 2 and Buzzer states as the beginning of the USB Ready sequence.

Step Index	1	2	3	4	5	6
LED 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
LED 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Buzzer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Duration (ms)	50	50	50	50	50	100

- To stop the buzzing, present a card to the reader and then remove it.
- Presenting a card will cause constant buzzing while the card is in the field. Buzzing will stop when the card is removed from the field.

### 5.1.3.2 Incorrect zero duration

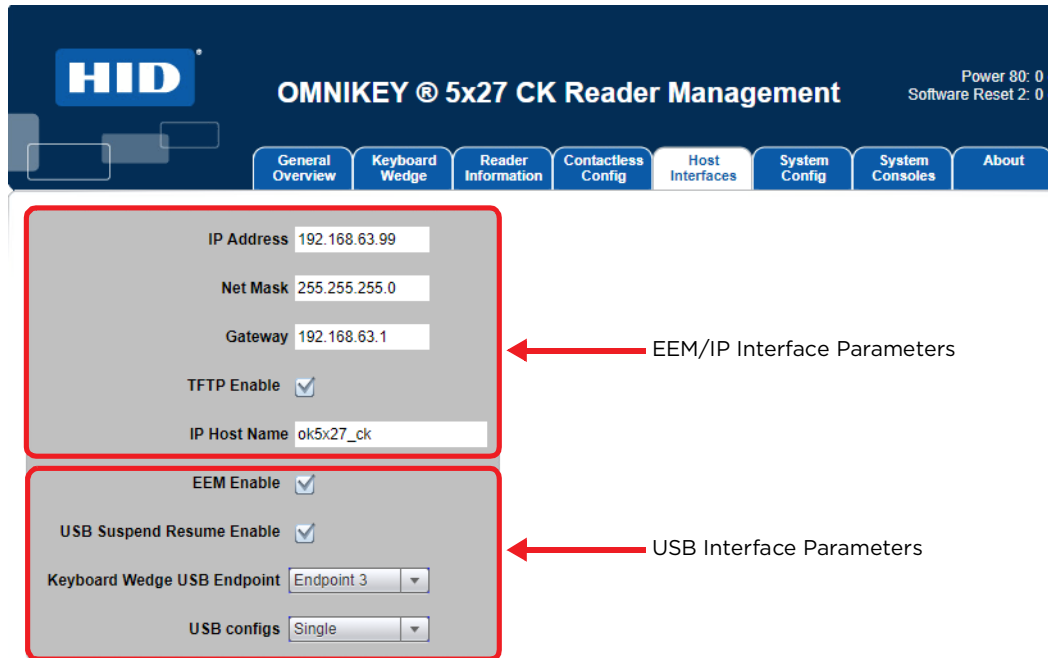
If a step's duration is set to 0, that step and all steps after it will not be considered. In the following example, both LEDs remain active, since step 2 has a zero duration:

Step Index	1	2	3	4	5	6
LED 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LED 2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Buzzer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Duration (ms)	100	0	0	0	0	0

## 5.2 Host interfaces

The OMNIKEY 5x27 supports multiple host interfaces including USB Endpoints. All the host interface options are manageable via the **Host Interfaces** tab.

### 5.3 Navigating the Host Interfaces tab



#### 5.3.1 EEM IP interface parameters

This section allows for the setup of Ethernet Interface parameters. It is suggested that documentation is maintained when changing these parameters.

The OMNIKEY 5x27 Default values are shown above.

**Note:** A configuration card can reset these settings to default if required.

##### IP Addressing

**IP Address, Net Mask** and **Gateway** are fully configurable. Once changed, the settings must be supported on the host PC to access the web based management tool. For instance, if the IP Address is changed to 192.168.63.100, this new setting must be entered as the new URL in the internet browser to access the management tool.



##### TFTP Enable

When **TFTP** is disabled, the TFTP capabilities of the reader are no longer allowed. For additional information on TFTP, see the *OMNIKEY 5x27CK Software Developer Guide* (5127-903).

##### IP Host Name

The IP hostname is configurable using the **IP Host Name** text box. The **IP Hostname** is limited to 15 characters in length.

### **EEM Enable**

When the **EEM Enable** option is selected, the OMNIKEY 5x27 will enumerate as a network adapter and the host/user may access the Web Based Management tool. When not selected (disabled), the Web Based management tool is not accessible.

## **5.3.2 USB Interface Parameters**

### **USB Suspend Resume Enable**

The **USB Suspend Resume Enable** option is not supported by all devices.

### **Keyboard Wedge USB Endpoint**

From the **Keyboard Wedge USB Endpoint** drop-down list, select one of the four USB endpoints that affect device enumeration and USB port transfers. These options and descriptions are available:

- Endpoint 0 - Control
- Endpoint 1 - Interrupt Transfers
- Endpoint 2 - Isochronous Transfers
- Endpoint 3 - Bulk Transfers

**Note:** The Keyboard Wedge USB Endpoint selected only affects the USB enumeration process when Keyboard Wedge is enabled. This is not a global parameter.

This page intentionally left blank.

# Chapter 6

## OMNIKEY® 5x27 configuration examples

### 6.1 Example 1 - Reading iCLASS® card PACS data

1. Enable **Keyboard Wedge** mode.
2. Select the **Keyboard Wedge** tab and select the **Card Data Selection** tab.
3. From the **Card Type** drop-down list, select **HID iCLASS**.
4. Select the **Enable HID iCLASS** option.
5. Select the **PACS** option.
6. On the **Out 2 Data Manipulation** tab, in the **PACS Prestrokes** field, enter **Start**.
7. Press Enter or change focus.

Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Book	Page	Block	Offset	Length
CSN	<input type="checkbox"/>		None							
PACS	<input checked="" type="checkbox"/>		None							
Custom 1	<input type="checkbox"/>			0	Kd	0	0	0	0	0
Custom 2	<input type="checkbox"/>			0	Kd	0	0	0	0	0
Custom 3	<input type="checkbox"/>			0	Kd	0	0	0	0	0

Datafield	String Padding	Prestrokes
CSN	<input type="checkbox"/> Char 0 Direction Leading Length 0	
PACS	<input type="checkbox"/> Char 0 Direction Leading Length 0	Start

8. Open a text editor and place the iCLASS Sample card into the RFID field over the antenna of the reader.
9. The Keyboard Wedge enters into the editor the word **Start** followed by the PACS data in hexadecimal format, for example: **Start07FFE00002**

## 6.2 Example 2 - Reading MIFARE card CSN

1. Go to the **Keyboard Wedge** tab and select the **Card Data Selection** tab.
2. From the **Card Type** drop-down menu, select **MIFARE Classic**.
3. Select the **Enable MIFARE Classic** option.
4. Select the **CSN** option.

Sets the Card Type									
Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Sector	Block	Offset	Length
CSN	<input checked="" type="checkbox"/>	None							
PACS	<input type="checkbox"/>								
Custom 1	<input type="checkbox"/>			0	A	0	0	0	0
Custom 2	<input type="checkbox"/>			0	A	0	0	0	0
Custom 3	<input type="checkbox"/>			0	A	0	0	0	0

5. Select the **Out 2 Data Manipulation** tab.
6. Enter **Start** into the **Prestrokes** field, and press Enter or change focus.
7. Enter **End** into the **Poststrokes** field, and press Enter or change focus.

Datafield	String Padding	Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char 0 Direction <b>Leading</b> Length 0	Start	End

8. Open a text editor and place the MIFARE 1k Sample card into the RFID field over the antenna of the reader.
9. The Keyboard wedge enters into the editor the word **Start** followed by the CSN data in hexadecimal format and the word **End**, for example: **Start7D1BF3AEEnd**



## 6.3 Example 3 - HID iCLASS PACS data filtering

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab.
3. From the **Card Type** drop-down menu, select **HID iCLASS**.
4. Select the **Enable HID iCLASS** option.
5. Select the **PACS** option.

Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Book	Page	Block	Offset	Length
CSN	<input type="checkbox"/>	None								
PACS	<input checked="" type="checkbox"/>	None								

6. Select the **Out 1 Data Manipulation** tab.
7. In the **PACS** row, perform the following steps:
  - a. In the **String Filtering** pane, select the checkbox and enter “f” in the **Char** field.
  - b. In the **String Format** pane, verify that HEX is selected.

Datafield	String Format	String Filtering	String Truncating
CSN	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0
PACS	HEX	<input checked="" type="checkbox"/> Char f Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0
Custom 1	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0

8. Select the **Out 2 Data Manipulation** tab.
9. Enter **<pacs>** into the **Prestrokes** text field, and press Enter or change focus.
10. Enter **</pacs>** into the **Poststrokes** text field, and press Enter or change focus.

Datafield	String Padding	Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char 0 Direction Leading Length 0		
PACS	<input type="checkbox"/> Char 0 Direction Leading Length 0	<pacs>	</pacs>

11. Open a text editor and place the iCLASS Sample card into the RFID field over the antenna of the reader.

12. The Keyboard Wedge enters into the editor the text <pacs> followed by the filtered PACS data in hexadecimal format followed by the text </pacs>. For example:

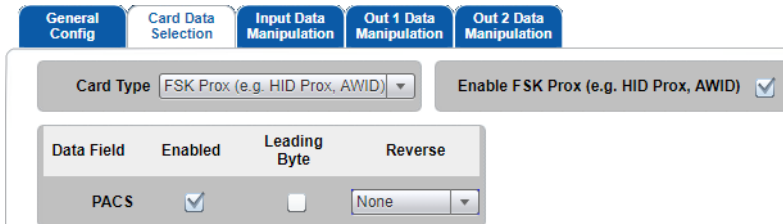
Raw PACS data (HEX): **6e1b500f9ff12e0**

KBW output: **<pacs>6e1b500912e0</pacs>**

**Note:** The character “f” has been filtered out.

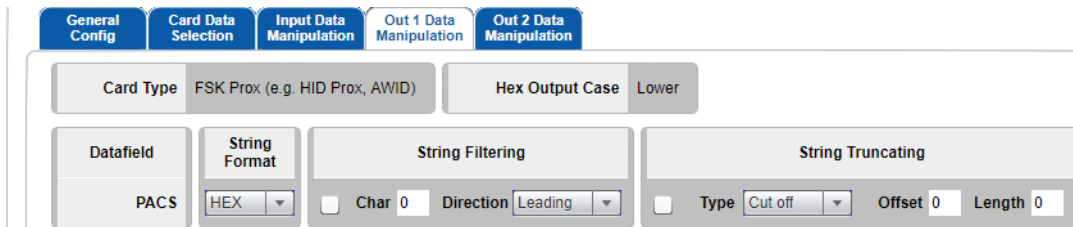
## 6.4 Example 4 - Prox card string padding

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab.
3. From the **Card Type** drop-down list, select **HID Prox**.
4. Select the **Enable HID Prox** option.
5. Select the **PACS** option.



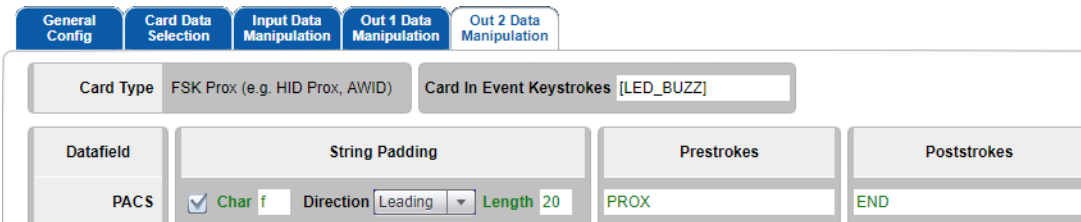
Data Field	Enabled	Leading Byte	Reverse
PACS	<input checked="" type="checkbox"/>	None	<input type="checkbox"/>

6. Select the **Out 1 Data Manipulation** tab.
7. In the **PACS** row **String Format** drop-down, select **HEX**.



Datafield	String Format	String Filtering	String Truncating
PACS	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0

8. Select the **Out 2 Data Manipulation** tab.
9. In the **PACS** row:
  - a. Select the **PACS** option in the **PACS** row.
  - b. Enter “f” in the **Char** field.
  - c. Select **Leading** in the **Direction** field.
  - d. Enter 20 in the **Length** field.
10. Enter **PROX** into the **Prestrokes** field, press Enter.
11. Enter **END** into the **Poststrokes** field, press Enter.



Datafield	String Padding	Prestrokes	Poststrokes
PACS	<input checked="" type="checkbox"/> Char f Direction Leading Length 20	PROX	END

12. Open a text editor and place an HID Prox card into the RFID field over the antenna of the reader:
  - If the data on the card is: **100000010000000001001111**
  - The output in the editor will be: **PROXfffffffff0202004fEND**

## 6.5 Example 5 - HID iCLASS, standard 26 bit, FC and CN

**Note:** Keyboard Wedge mode is already enabled

**Note:** In order to read the Access Control Data, you have to know the format in which the card was configured and its data structure.

Using a Wiegand 26 bit format, which has the following structure:

PAAAAAAAAABBBBBBBBBBBBBBBBBBP

Where:

P - parity

A - facility code (FC)

B - card number (CN)

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab.
3. From the **Card Type** drop-down, select **HID iCLASS**.
4. Select the **Enable HID iCLASS** option.
5. Configure the PACS Custom field as shown:
  - a. Enable **PACS Custom** and enter the offset and length:

Data Field	Enabled	Reverse	Offset (Bytes)	Length (Bytes)
CSN Custom 1	<input type="checkbox"/>	None	0	0
CSN Custom 2	<input type="checkbox"/>	None	0	0

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	8
PACS Custom 2	<input type="checkbox"/>	None	9	16
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

b. Select string format for Custom PACS:

The screenshot shows a configuration window with tabs for General Config, Card Data Selection, Input Data Manipulation, Out 1 Data Manipulation, and Out 2 Data Manipulation. The 'Input Data Manipulation' tab is active. It displays settings for Custom 3, Custom CSN 1, Custom CSN 2, Custom PACS 1, and Custom PACS 2. For each entry, there are sections for String Format, String Filtering, and String Truncating. The 'String Format' dropdowns for Custom PACS 1 and Custom PACS 2 are highlighted with a red box and set to 'DEC'.

c. Enter pre- and post-strokes:

The screenshot shows the same configuration window as above, but now the 'String Padding', 'Prestrokes', and 'Poststrokes' sections are visible. The 'Prestrokes' and 'Poststrokes' fields for Custom PACS 1 and Custom PACS 2 are highlighted with a red box. Custom PACS 1 has 'FC =' in the Prestrokes field and an empty Poststrokes field. Custom PACS 2 has 'CN =' in the Prestrokes field and an empty Poststrokes field.

6. Open a text editor and place an HID iCLASS card into the RFID field over the antenna of the reader. The data must be written similarly to the example below:

FC = 10\_\_CN = 5723\_\_

## 6.6 Example 6 – PIV 75 bit card number

**Note:** Keyboard Wedge mode is already enabled.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab.
3. From the **Card Type** drop-down, select PIV.
4. Select the **Enable PIV** option.
5. Enable the **FASC\_N Custom 1** option.

Data Field	Enabled	Reverse	Offset (Bytes)	Length (Bytes)
CSN Custom 1	<input type="checkbox"/>	None	0	0
CSN Custom 2	<input type="checkbox"/>	None	0	0

Data Field	Enabled	Remove Parity	Reverse BCD	Reverse	Offset (Bits)	Length (Bits)
FASC_N Custom 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	None	44	24

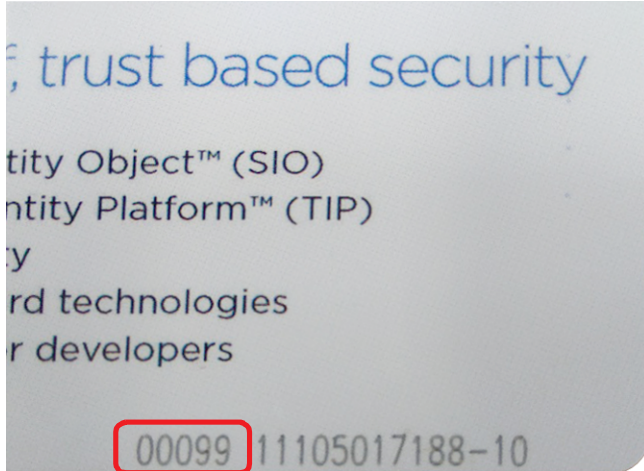
6. Enter 44 in the **Offset (Bits)** field.
7. Enter 24 in the **Length (Bits)** field.
8. Select the **Out 2 Data Manipulation** tab.

Datafield	String Padding	Prestrokes	Poststrokes
FASC-N Cust 1	<input type="checkbox"/> Char 0 Direction Leading Length 0	CredNum:	[ENTER]

9. Enter your required pre-strokes and post-strokes for **FASC\_N Cust 1**.
10. Open a text editor and place a PIV card on the reader. The desired data is written to the text editor.

## 6.7 HID PROX 26-bit format H10301 facility code and user ID (decimal output).

In this example, the number on the card is 99 (decimal).



1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type: FSK Prox (e.g. HID Prox, AWID)		Enable FSK Prox (e.g. HID Prox, AWID) <input checked="" type="checkbox"/>		
Data Field	Enabled	Leading Byte	Reverse	
PACS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	None	
Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	8
PACS Custom 2	<input type="checkbox"/>	None	9	16
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

**Note:** PACS is set to show only for demonstration purposes.

3. Select the **Out 1 Data Manipulation** tab and make the following settings:

**Note:** PACS data is set to binary only for demonstration purposes.

4. Select the **Out 2 Data Manipulation** tab and make the following settings:

This will produce the output:

```
PACS 10000000100000000011000111
Facility Code 1
Card Data 99
```

The binary PACS can be interpreted as shown below, to obtain the facility code 1 and data 99 (decimal), where parity bits are purple, facility code is blue, and ID number is red:

```
Bit positions: 0 1      8 9              24 25
PACS:         1 00000001 000000001100011 1
```

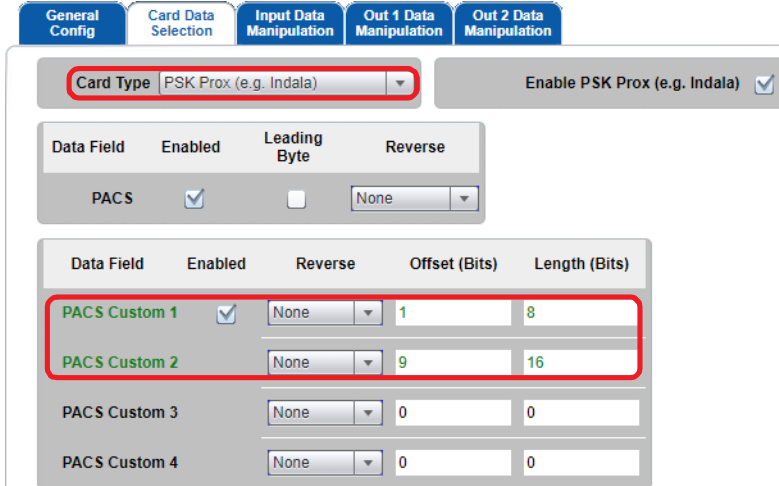
Binary 01100011 = hex 63 = decimal 99.

**Note:** The same formatting can be used with iCLASS, iCLASS Seos® and other cards using the H10301 format.



## 6.8 Indala® PROX default format

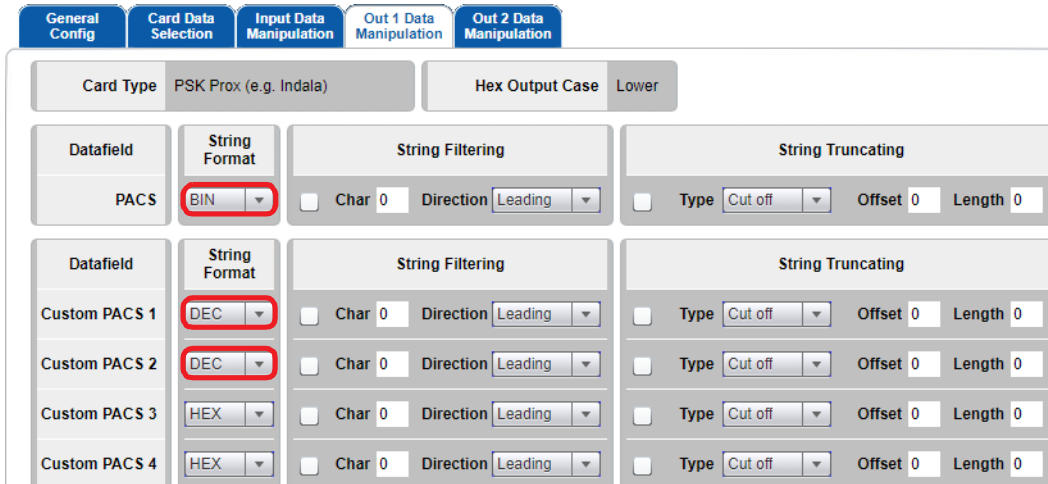
1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



Card Type	PSK Prox (e.g. Indala)	Enable PSK Prox (e.g. Indala)																									
<table border="1"> <thead> <tr> <th>Data Field</th> <th>Enabled</th> <th>Leading Byte</th> <th>Reverse</th> </tr> </thead> <tbody> <tr> <td>PACS</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td>None</td> </tr> </tbody> </table>	Data Field	Enabled	Leading Byte	Reverse	PACS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	None																			
Data Field	Enabled	Leading Byte	Reverse																								
PACS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	None																								
<table border="1"> <thead> <tr> <th>Data Field</th> <th>Enabled</th> <th>Reverse</th> <th>Offset (Bits)</th> <th>Length (Bits)</th> </tr> </thead> <tbody> <tr> <td>PACS Custom 1</td> <td><input checked="" type="checkbox"/></td> <td>None</td> <td>1</td> <td>8</td> </tr> <tr> <td>PACS Custom 2</td> <td><input type="checkbox"/></td> <td>None</td> <td>9</td> <td>16</td> </tr> <tr> <td>PACS Custom 3</td> <td><input type="checkbox"/></td> <td>None</td> <td>0</td> <td>0</td> </tr> <tr> <td>PACS Custom 4</td> <td><input type="checkbox"/></td> <td>None</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)	PACS Custom 1	<input checked="" type="checkbox"/>	None	1	8	PACS Custom 2	<input type="checkbox"/>	None	9	16	PACS Custom 3	<input type="checkbox"/>	None	0	0	PACS Custom 4	<input type="checkbox"/>	None	0	0		
Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)																							
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	8																							
PACS Custom 2	<input type="checkbox"/>	None	9	16																							
PACS Custom 3	<input type="checkbox"/>	None	0	0																							
PACS Custom 4	<input type="checkbox"/>	None	0	0																							

**Note:** PACS is set to show only for demonstration purposes.

3. Select the **Out 1 Data Manipulation** tab and make the following settings:



Card Type	PSK Prox (e.g. Indala)	Hex Output Case	Lower																								
<table border="1"> <thead> <tr> <th>Datafield</th> <th>String Format</th> <th>String Filtering</th> <th>String Truncating</th> </tr> </thead> <tbody> <tr> <td>PACS</td> <td>BIN</td> <td><input type="checkbox"/> Char 0 Direction Leading</td> <td><input type="checkbox"/> Type Cut off Offset 0 Length 0</td> </tr> <tr> <td>Custom PACS 1</td> <td>DEC</td> <td><input type="checkbox"/> Char 0 Direction Leading</td> <td><input type="checkbox"/> Type Cut off Offset 0 Length 0</td> </tr> <tr> <td>Custom PACS 2</td> <td>DEC</td> <td><input type="checkbox"/> Char 0 Direction Leading</td> <td><input type="checkbox"/> Type Cut off Offset 0 Length 0</td> </tr> <tr> <td>Custom PACS 3</td> <td>HEX</td> <td><input type="checkbox"/> Char 0 Direction Leading</td> <td><input type="checkbox"/> Type Cut off Offset 0 Length 0</td> </tr> <tr> <td>Custom PACS 4</td> <td>HEX</td> <td><input type="checkbox"/> Char 0 Direction Leading</td> <td><input type="checkbox"/> Type Cut off Offset 0 Length 0</td> </tr> </tbody> </table>	Datafield	String Format	String Filtering	String Truncating	PACS	BIN	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0	Custom PACS 1	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0	Custom PACS 2	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0	Custom PACS 3	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0	Custom PACS 4	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0			
Datafield	String Format	String Filtering	String Truncating																								
PACS	BIN	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0																								
Custom PACS 1	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0																								
Custom PACS 2	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0																								
Custom PACS 3	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0																								
Custom PACS 4	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0																								

**Note:** PACS data is set to binary only for demonstration purposes.

4. Select the **Out 2 Data Manipulation** tab and make the following settings:

This will produce the output:

```
PACS 10000000100000000000010011
Facility Code 1
Card Data 9
```

The binary PACS can be interpreted as shown below, to obtain the facility code 1 and data 9 (decimal), where parity bits are purple, facility code is blue, and ID number is red:

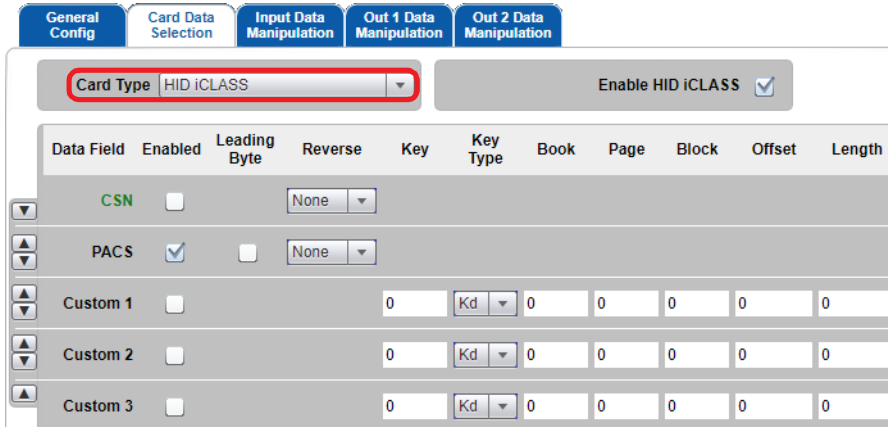
```
Bit positions: 0 1      8 9              24 25
PACS:         1 00000001 0000000000001001 1
```

Binary 10011 = hex 09 = decimal 9.

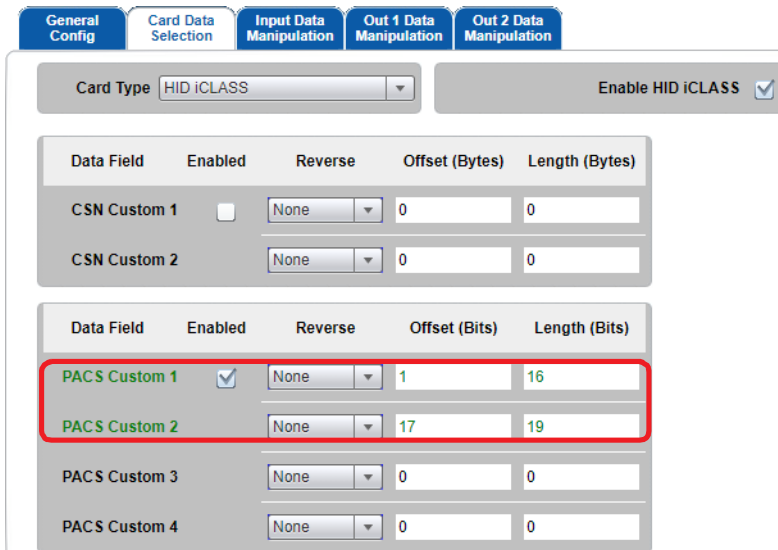
## 6.9 iCLASS H10304 format facility code and user ID (decimal output)

In this example, the number on the card is 1.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Book	Page	Block	Offset	Length
CSN	<input type="checkbox"/>		None							
PACS	<input checked="" type="checkbox"/>		None							
Custom 1	<input type="checkbox"/>			0	Kd	0	0	0	0	0
Custom 2	<input type="checkbox"/>			0	Kd	0	0	0	0	0
Custom 3	<input type="checkbox"/>			0	Kd	0	0	0	0	0



Data Field	Enabled	Reverse	Offset (Bytes)	Length (Bytes)
CSN Custom 1	<input type="checkbox"/>	None	0	0
CSN Custom 2	<input type="checkbox"/>	None	0	0

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	16
PACS Custom 2	<input type="checkbox"/>	None	17	19
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

**Note:** PACS is set to show only for demonstration purposes.

3. Select the **Out 1 Data Manipulation** tab and make the following settings:

**Note:** PACS data is set to binary only for demonstration purposes.

4. Select the **Out 2 Data Manipulation** tab and make the following settings:

This will produce the output:

```
PACS 1000000000000000010000000000000000010  
Facility Code 1  
Card Number 1
```

The binary PACS can be interpreted as shown below, to obtain the facility code 1 and data 1 (decimal), where parity bits are **purple**, facility code is **blue**, and card number is **red**:

```
Bit positions: 1 2                17 18                36 37  
PACS:         1 0000000000000001 0000000000000001 0
```

**Note:** The same formatting can be used with HID Prox or other cards using the H10301 format.

### 6.10 MIFARE Classic 26-bit format facility code and user ID (decimal output)

In this example, the number on the card is 90 (decimal).



1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:

Data Field	Enabled	Leading Byte	Reverse	Key	Key Type	Sector	Block	Offset	Length
CSN	<input type="checkbox"/>		None						
PACS	<input checked="" type="checkbox"/>		None						
Custom 1	<input type="checkbox"/>			0	A	0	0	0	0
Custom 2	<input type="checkbox"/>			0	A	0	0	0	0
Custom 3	<input type="checkbox"/>			0	A	0	0	0	0

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	8
PACS Custom 2	<input type="checkbox"/>	None	9	16
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

**Note:** PACS is set to show only for demonstration purposes.

3. Select the **Out 1 Data Manipulation** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type: MIFARE Classic		Hex Output Case: Lower		
Datafield	String Format	String Filtering		String Truncating
CSN	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
PACS	BIN	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom 1	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom 2	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom 3	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Datafield	String Format	String Filtering		String Truncating
Custom PACS 1	DEC	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom PACS 2	DEC	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom PACS 3	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0
Custom PACS 4	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off    Offset: 0    Length: 0

**Note:** PACS data is set to binary only for demonstration purposes.

This will produce the output:

```
PACS 10000000100000000010110101
Facility Code 1
Card Number 90
```

The binary PACS can be interpreted as shown below, to obtain the facility code 1 and data 90 (decimal), where parity bits are purple, facility code is blue, and card number is red:

```
Bit positions: 0 1            8 9                    24 25
PACS:            1 0000001 000000001011010 1
```

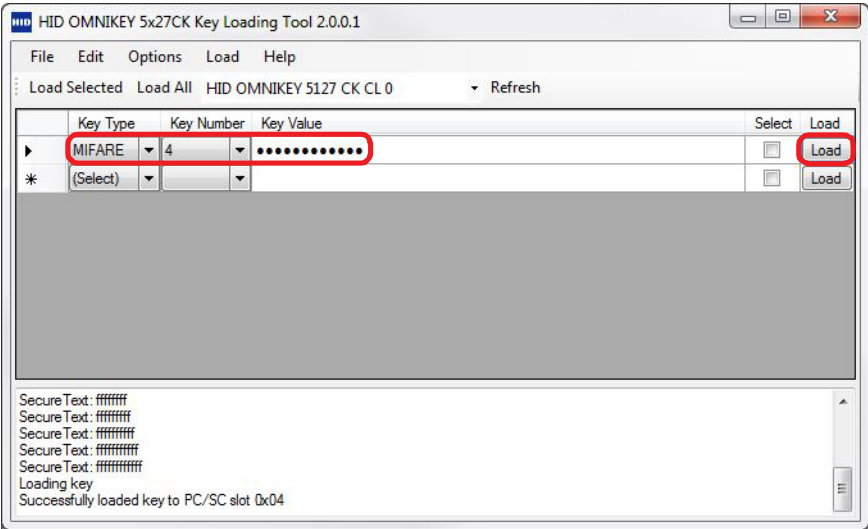
Binary 1011010 = hex 5A = decimal 90.

### 6.11 MIFARE Classic sector read, including load keys:

To perform a sector read from a MIFARE Classic card, you must start by loading the read key for the required sector into the reader. This is best achieved using the HID Key Loading tool, which is freely available on the HID Developer Center.

**Note:** To load keys, the reader must be in CCID mode.

The screen shot below shows the key loading tool being used to set a key of 0xFFFFFFFFFFFF into key slot number 4.

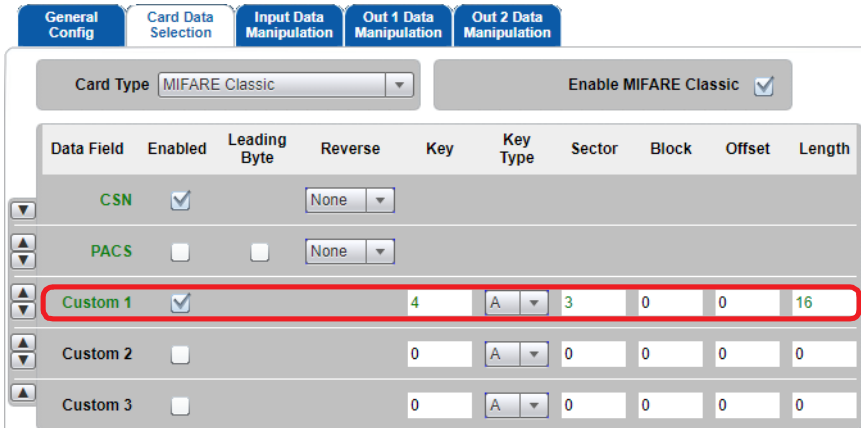


- Click **Load** to load the key into the reader.

You can now switch the reader into Keyboard Wedge mode.

The following example reads custom data from all 16 bytes of sector 3, block 0, using key number 4 as set up above.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



**Note:** CSN is set to show only for demonstration purposes, and is not necessary for reading custom data.



3. Select the **Out 1 Data Manipulation** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type	MIFARE Classic	Hex Output Case	Lower	
Datafield	String Format	String Filtering		String Truncating
CSN	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off, Offset: 0, Length: 0
PACS	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off, Offset: 0, Length: 0
Custom 1	ASCII	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off, Offset: 0, Length: 0
Custom 2	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off, Offset: 0, Length: 0
Custom 3	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type: Cut off, Offset: 0, Length: 0

4. Select the **Out 2 Data Manipulation** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type	MIFARE Classic	Card In Event Keystrokes	[LED_BUZZ]	
Datafield	String Padding		Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char 0	Direction: Leading, Length: 0	CSN	[ENTER]
PACS	<input type="checkbox"/> Char 0	Direction: Leading, Length: 0		
Custom 1	<input type="checkbox"/> Char 0	Direction: Leading, Length: 0	Custom Data:	[ENTER]
Custom 2	<input type="checkbox"/> Char 0	Direction: Leading, Length: 0		
Custom 3	<input type="checkbox"/> Char 0	Direction: Leading, Length: 0		

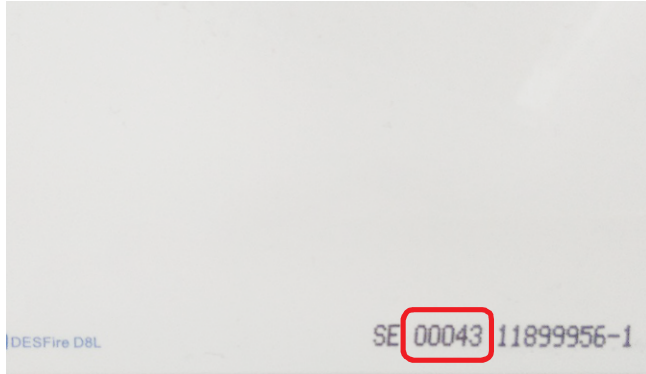
This will produce the output:

```
CSN fbc157e1
```

```
Custom Data: HID Mifare card!
```

## 6.12 MIFARE DESFire H10302 format, user ID (decimal output):

In this example, the number on the card is 43 (decimal).



1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:

Data Field	Enabled	Leading Byte	Reverse	App ID	File Num	Start Len	Card Rdr Key	Auth	File Type	File Comms	Encryption	AV1 Diversify	
CSN	<input type="checkbox"/>		None										
PACS	<input checked="" type="checkbox"/>		None										
Custom 1	<input type="checkbox"/>		1	0	0	0	0	240	<input type="checkbox"/>	Standard	None	DES/3DES	None
Custom 2	<input type="checkbox"/>		1	0	0	0	0	240	<input type="checkbox"/>	Standard	None	DES/3DES	None
Custom 3	<input type="checkbox"/>		1	0	0	0	0	240	<input type="checkbox"/>	Standard	None	DES/3DES	None

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	1	35
PACS Custom 2	<input type="checkbox"/>	None	0	0
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

**Note:** PACS is set to show only for demonstration purposes.



3. Select the **Out 1 Data Manipulation** tab and make the following settings:

General Config		Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation	
Card Type		MIFARE Desfire EV1			Hex Output Case	Lower
Datafield	String Format	String Filtering		String Truncating		
CSN	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0	
PACS	BIN	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0	
Custom 1	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0	
Custom 2	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0	
Custom 3	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0	

Datafield	String Format	String Filtering		String Truncating	
Custom PACS 1	DEC	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0
Custom PACS 2	HEX	<input type="checkbox"/> Char 0	Direction: Leading	<input type="checkbox"/> Type	Cut off, Offset 0, Length 0

**Note:** PACS data is set to binary only for demonstration purposes.

4. Select the **Out 2 Data Manipulation** tab and make the following settings:

General Config		Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation	
Card Type		MIFARE Desfire EV1			Card In Event Keystrokes	[LED_BUZZ]
Datafield	String Padding		Prestrokes	Poststrokes		
CSN	<input type="checkbox"/> Char 0	Direction: Leading, Length 0				
PACS	<input type="checkbox"/> Char 0	Direction: Leading, Length 0	PACS	[ENTER]		
Custom 1	<input type="checkbox"/> Char 0	Direction: Leading, Length 0				
Custom 2	<input type="checkbox"/> Char 0	Direction: Leading, Length 0				
Custom 3	<input type="checkbox"/> Char 0	Direction: Leading, Length 0				

Datafield	String Padding		Prestrokes	Poststrokes	
Custom PACS 1	<input type="checkbox"/> Char 0	Direction: Leading, Length 0	Card Number	[ENTER]	
Custom PACS 2	<input type="checkbox"/> Char 0	Direction: Leading, Length 0			
Custom PACS 3	<input type="checkbox"/> Char 0	Direction: Leading, Length 0			
Custom PACS 4	<input type="checkbox"/> Char 0	Direction: Leading, Length 0			

This will produce the output:

```
PACS 00000000000000000000000000000001010111
Card Number 43
```

The binary PACS can be interpreted as shown below, to obtain the facility data 43 (decimal), where parity bits are purple, facility code is not required, and card number is red:

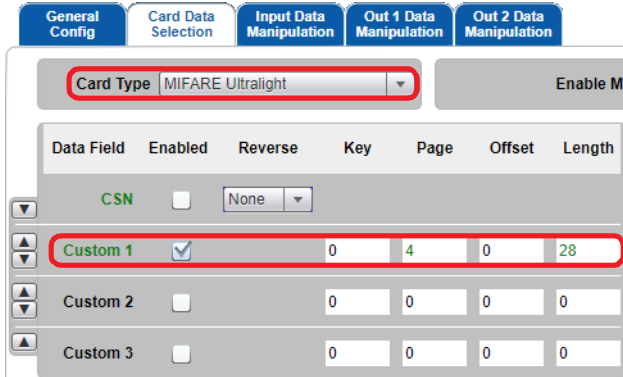
Bit positions: 1 2 36 37  
PACS: 0 00000000000000000000000000000101011 1

Binary 00101011 = hex 2B = decimal 43.

## 6.13 MIFARE Ultralight sector read, including load keys

MIFARE Ultralight does not use keys, so there is no need to load any keys. However, the keys section is present for Ultralight C, which does use keys.

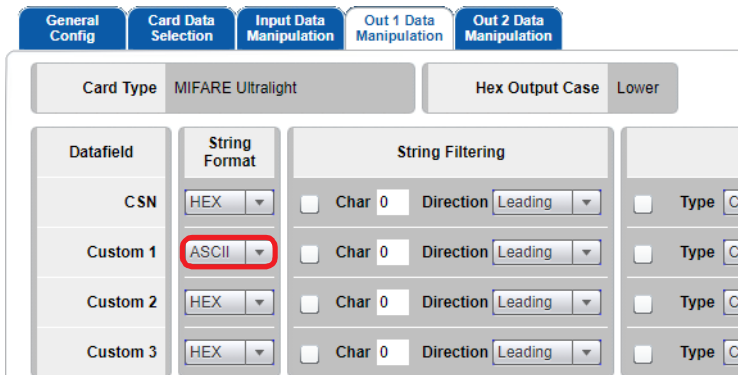
1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



Data Field	Enabled	Reverse	Key	Page	Offset	Length
CSN	<input type="checkbox"/>	None				
Custom 1	<input checked="" type="checkbox"/>		0	4	0	28
Custom 2	<input type="checkbox"/>		0	0	0	0
Custom 3	<input type="checkbox"/>		0	0	0	0

**Note:** Although Ultralight pages are only 4 bytes long, you can specify any length, as the reader will read subsequent pages until it has enough bytes.

3. Select the **Out 1 Data Manipulation** tab and make the following settings:



Datafield	String Format	String Filtering	Type
CSN	HEX	<input type="checkbox"/> Char 0 Direction Leading	Cu
Custom 1	ASCII	<input type="checkbox"/> Char 0 Direction Leading	Cu
Custom 2	HEX	<input type="checkbox"/> Char 0 Direction Leading	Cu
Custom 3	HEX	<input type="checkbox"/> Char 0 Direction Leading	Cu

This will produce the output:

```
HID Global Mifare Ultralight
```

**Note:** This 28 bytes of data is saved over 7 pages of 4 bytes each.

## 6.14 FeliCa CSN with HEX and DEC output:

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:

Data Field	Enabled	Reverse	Service code	Block	Offset	Length
CSN	<input checked="" type="checkbox"/>	None				
Custom 1	<input type="checkbox"/>		0	0	0	0
Custom 2	<input type="checkbox"/>		0	0	0	0
Custom 3	<input type="checkbox"/>		0	0	0	0

3. Select the **Out 1 Data Manipulation** tab. To see the FeliCa IDm (CSN), select HEX/DEC/ASCII depending on the required output:

Datafield	String Format	String Filtering
CSN	HEX	<input type="checkbox"/> Char 0 Direction Leading
Custom 1	HEX	<input type="checkbox"/> Char 0 Direction Leading
Custom 2	HEX	<input type="checkbox"/> Char 0 Direction Leading
Custom 3	HEX	<input type="checkbox"/> Char 0 Direction Leading

Typical card output (HEX and DEC):

011303000b174d07

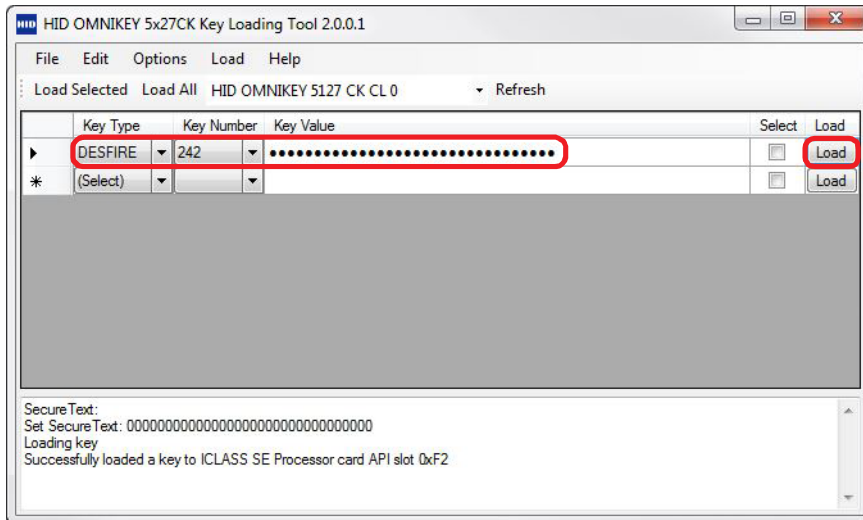
77408917316390151

## 6.15 MIFARE DESFire custom application read and loading keys:

A DESFire key must be loaded into the reader. This is best achieved using the HID Key Loading tool, which is freely available on the HID Developer Center.

**Note:** To load keys, the reader must be in CCID mode.

The screen shot below shows the key loading tool being used to set a key of 0x00000000000000000000000000000000 into key slot number 242.

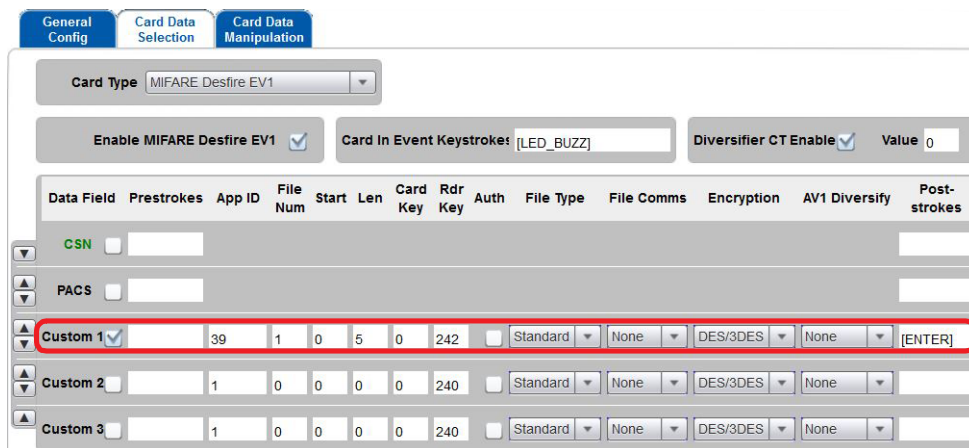


- Click **Load** to load the key into the reader.

You can now switch the reader into Keyboard Wedge mode.

The following example reads custom data 5 bytes long from application ID 0x27 (39 decimal), file ID 0x01, using key 242 as set up above.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



Since the data on the card is in hex, the other settings in the **Card Data Manipulation** tab can remain at their defaults. On this sample card this produces the output:

```
0123456780
```

## 6.16 Seos credentials, corporate 1000 format, facility code and user ID (decimal output)

In this example, the Seos card data is:

- Corporate 1000 format
- Facility Code: 0x000FFF (4095 decimal)
- User ID/Card Number: 0x000001 (1 decimal)

To read the facility code and card number, it is necessary to configure Keyboard Wedge as follows.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:

Data Field	Enabled	Leading Byte	Reverse	ADF Oid	Priv Enc Key OID	Priv MAC Key OID	Auth Key OID	Tag	Offset	Length
CSN	<input type="checkbox"/>		None							
PACS	<input checked="" type="checkbox"/>		None							
Custom 1	<input type="checkbox"/>			0	0	0	0	0	0	0
Custom 2	<input type="checkbox"/>			0	0	0	0	0	0	0
Custom 3	<input type="checkbox"/>			0	0	0	0	0	0	0

Data Field	Enabled	Reverse	Offset (Bits)	Length (Bits)
PACS Custom 1	<input checked="" type="checkbox"/>	None	2	22
PACS Custom 2	<input type="checkbox"/>	None	24	23
PACS Custom 3	<input type="checkbox"/>	None	0	0
PACS Custom 4	<input type="checkbox"/>	None	0	0

3. Select the **Out 1 Data Manipulation** tab and make the following settings:

Card Type	Hex Output Case
Seos	Lower

Datafield	String Format	String Filtering	String Truncating
CSN	HEX	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0
PACS	BIN	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0

Datafield	String Format	String Filtering	String Truncating
Custom PACS 1	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0
Custom PACS 2	DEC	<input type="checkbox"/> Char 0 Direction Leading	<input type="checkbox"/> Type Cut off Offset 0 Length 0



4. Select the **Out 2 Data Manipulation** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type: Seos		Card In Event Keystrokes: [LED_BUZZ]		
Datafield	String Padding		Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	
PACS	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	PACS: [ENTER]
Datafield	String Padding		Prestrokes	Poststrokes
Custom PACS 1	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	Facility Code: [ENTER]
Custom PACS 2	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	Card Number: [ENTER]
Custom PACS 3	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	
Custom PACS 4	<input type="checkbox"/> Char 0	Direction: Leading	Length: 0	

This will produce the output:

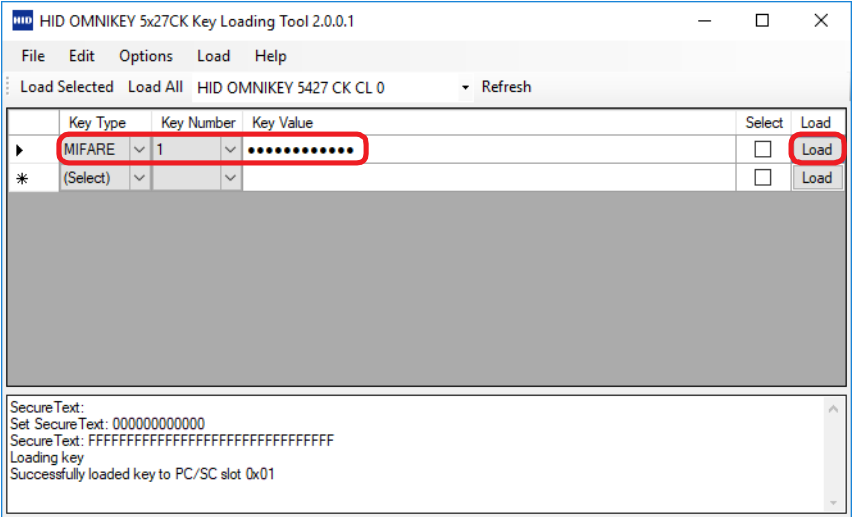
```
PACS: 0100000000001111111111110000000000000000000000011
Facility Code: 4095
Card Number: 1
```

### 6.17 MIFARE Plus custom sector read with load keys

To be able to read data from MIFARE Plus card (Security Level 3 is now the only one supported by Keyboard Wedge) it is necessary to load a key for authentication. This is best achieved using the HID Key Loading tool, which is freely available on the HID Developer Center. It is also possible to send PC/SC command Load Key; refer to *OMNIKEY 5x27CK Software Developer Guide* (5127-903).

**Note:** To load keys, the reader must be in CCID mode.

The screen shot below shows the key loading tool being used to set a key of 0xFFFFFFFFFFFFFFFFFFFFFFFF into key slot number 1.

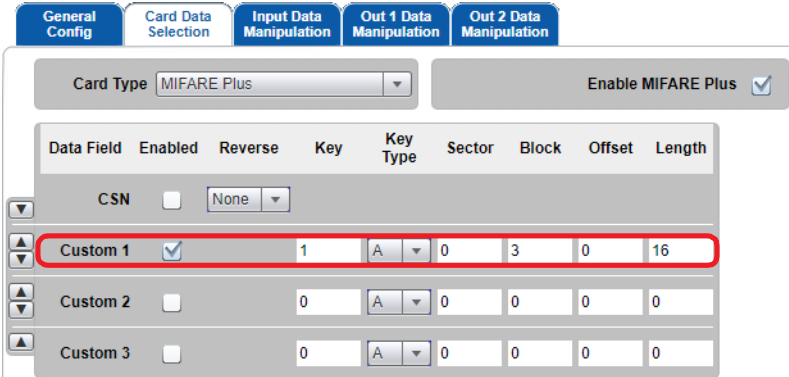


- Click **Load** to load the key into the reader.

You can now switch the reader into Keyboard Wedge mode.

The following example reads custom data from the card.

1. Select the **Keyboard Wedge** tab.
2. Select the **Card Data Selection** tab and make the following settings:



3. Select the **Out 2 Data Manipulation** tab and make the following settings:

General Config	Card Data Selection	Input Data Manipulation	Out 1 Data Manipulation	Out 2 Data Manipulation
Card Type	MIFARE Plus		Card In Event Keystrokes [LED_BUZZ]	
Datafield	String Padding		Prestrokes	Poststrokes
CSN	<input type="checkbox"/> Char 0	Direction <span>Leading</span> Length 0		
Custom 1	<input type="checkbox"/> Char 0	Direction <span>Leading</span> Length 0	Custom Data 1:	[ENTER]

This allows all data from block 3 to be read from the card, and produces the following output:

```
Custom Data: 000000000000ffff078069ffffffffffff
```

This page intentionally left blank.

## Description of fields

---

**Note:** All numerical data fields (e.g. Length, Offset, File Num etc.) can be entered either in decimal or in hex. Hex value needs to be prefixed with “0x”.

### A.1 Enable Card Type

All card types have this option. It enables the keyboard wedge for the relevant card type. If not enabled, the keyboard wedge will not try to process the card as a Seos® card. It may however still try to process it as another card type if it fits more than one type. For example, a MIFARE Classic card could also be processed as a ISO14443A card if the Generic ISO14443A card type is enabled. If you wish to block the processing of the particular card type, then leave it enabled, but disable all of its data fields.

### A.2 Card In Event Keystrokes

**Note:** From firmware version 01.02.00f7, this field has moved from **Card Data Selection** to the **Out 2 Data Manipulation** tab.

All card types have this field. These keystrokes will be sent before outputting any other keyboard wedge data for this card type, even if all other fields are disabled.

### A.3 Pre-strokes

**Note:** From firmware version 01.02.00f7, this field has moved from **Card Data Selection** to the **Out 2 Data Manipulation** tab.

There is a pre-strokes setting for every keyboard wedge data field (e.g. CSN, PACS, custom data, etc.). These keystrokes are sent before outputting the data for each field. By default, pre-strokes will not be output if an error occurs reading a field. However, this is not the case if the **Tech Order** option is disabled and the **Allow Pre-strokes and Post-strokes for Errors option** is enabled.

### A.4 Post-strokes

**Note:** From firmware version 01.02.00f7, this field has moved from **Card Data Selection** to the **Out 2 Data Manipulation** tab.

These behave the same as pre-strokes, except that they are output after the data for the relevant field rather than before.

## A.5 CSN

This option enables the outputting of the serial number obtained during anti-collision.

## A.6 CSN Custom

This option enables the outputting of the serial number obtained during anti-collision.

### A.6.1 Reverse

This option allows reversing of CSN data. There are two different reverse options: byte and binary. Each is described in *Section 3.6: Input Data Manipulation tab*.

### A.6.2 Offset

The offset specifies the position, in bits, within the CSN data from which to start outputting data. Any value between zero and the length of the CSN data is allowed.

### A.6.3 Length

The length option specifies the number of bits of CSN data, starting from the offset value, to output.

## A.7 PACS

This option is enabled only for card types which may contain HID PACS data (MIFARE Classic, iCLASS®, Seos®, BLE Seos, MIFARE DESFire EV1) and LF card types. The option enables the output of the whole of the PACS contained on the card.

## A.8 PACS Custom

This option is enabled only for card types which may contain HID PACS data (MIFARE Classic, Prox, iCLASS, Seos, BLE Seos, MIFARE DESFire EV1). If enabled, it allows individual parts of the reader PACS data to be output. Up to four custom fields are provided and each one can be specified independently.

### A.8.1 Offset

The offset specifies the position in the PACS data, in bits, from which to start outputting data. Any value between zero and the length of the PACS data is allowed.

### A.8.2 Length

The length option specifies the number of bits of PACS data, starting from the offset value, to output.

## A.9 iCLASS Custom Fields

Each iCLASS custom has the options listed below.

### A.9.1 Key

This is the number of the key slot that the key was loaded to in order to authenticate to the card. Although any value in the range 0-255 will be accepted, the reader normally expects iCLASS keys to be loaded to slots in the range 33-52.

### A.9.2 Key Type

This can be Kd or Kc. This specifies the type of iCLASS key used to authenticate to the card. The choice of Kd or Kc depends on the page application limit set in the configuration block of the page being authenticated. Use Kd to authenticate to the area before the application limit and Kc to authenticate to the area after. Refer to the Picopass datasheets for more information.

### A.9.3 Book

This is the book address of the iCLASS card to read. The only valid value for 2KS and 16KS cards is zero. For 32KS cards, the value can be zero or one.

### A.9.4 Page

This is the page address of the chosen iCLASS book from which to start reading:

- For 2KS cards, or books of 16KS or 32KS cards configured with a single page per book, the only valid value is zero.
- If the book is configured with multiple pages per book, then the valid values are zero to seven.

### A.9.5 Block

The block option specifies the block of the page to start reading data from. For 2K pages the valid values are 0 to 31 and for 16K pages the valid values are 0 to 255.

### A.9.6 Offset

The offset specifies the position within the block, in bytes, at which to start reading the data. Although the size of an iCLASS block is eight bytes, values in the range 0 to 255 bytes are accepted. If the offset is greater than the size of the block, then the keyboard wedge will move into the following blocks until the offset has been reached.

### A.9.7 Length

The length specifies the number of bytes to read from the card. The maximum allowed length is 255 bytes. If the number of bytes is greater than the block size (8 bytes), then the keyboard wedge will continue to read the following blocks until the correct number of bytes have been read. However, the keyboard wedge will not be able to continue if the end of the application is reached, as a different key will be needed to authenticate.

## A.10 MIFARE Classic and MIFARE Plus Custom Fields

### A.10.1 Key

This is the number of the key slot that the key was loaded to in order to authenticate to the card. Although any value in the range 0-255 will be accepted, the reader normally expects MIFARE Classic keys to be loaded to slots in the range 0-31.

### A.10.2 Key Type

This can be either type A or type B. This specifies the type of MIFARE key used to authenticate to the card. The choice of type A or type B depends on the access conditions in the sector trailer of the sector being authenticated. Key type A is the most commonly used key type for card reads.

### A.10.3 Sector

This is the sector address of the MIFARE Classic card to read:

- For MIFARE Classic 1K, the sector value can be between 0 and 15, inclusive.
- For MIFARE Plus 2K, cards can have sector values from 0 up to and including 31.
- For MIFARE 4K, the value can be anything up to and including 39.

### A.10.4 Block

The block option specifies the block of the sector to start reading data from:

- For sector values up to and including fifteen, the block can be anything from zero up to and including three.
- For sectors greater than fifteen, the blocks can be anything from 0 up to and including fifteen.

The block value range is card dependent; please refer to MIFARE card specifications.

### A.10.5 Offset

The offset specifies the position within the block, in bytes, at which to start reading the data. Although the size of a MIFARE block is sixteen bytes, values in the range 0 to 255 bytes are accepted. If the offset is greater than the size of the block then the keyboard wedge will move into the following blocks until the offset has been reached.

### A.10.6 Length

The length specifies the number of bytes to read from the card. The maximum allowed length is 255 bytes. If the number of bytes is greater than the block size (16 bytes), then the keyboard wedge will continue to read the following blocks until the correct number of bytes have been read. However, the keyboard wedge will not be able to continue if the end of the sector is reached, as a different key will be needed to authenticate.



## A.11 MIFARE Ultralight Custom Fields

### A.11.1 Key

This is the number of the key slot that the key was loaded to in order to authenticate to the card. If authentication is not required, this value can be ignored. Although any value in the range 0-255 will be accepted, the reader normally expects MIFARE Ultralight keys to be loaded to slots in the range 240-255.

### A.11.2 Page

The page option specifies the page to start reading data from:

- For standard Ultralight, the page can be in the range 0 to 15.
- For Ultralight C, the page value can be up to and including 39.

**Note:** Although Ultralight C memory continues up to page 47, the remaining pages are not readable.

The value can be entered either in decimal or in hex (by placing "0x" before the hex value).

### A.11.3 Offset

The offset specifies the position within the page, in bytes, at which to start reading the data. Although the size of an Ultralight page is four bytes, values in the range 0 to 255 bytes are accepted. If the offset is greater than the size of the page, the keyboard wedge will move into the following pages until the offset has been reached.

### A.11.4 Length

The length specifies the number of bytes to read from the card. The maximum allowed length is 255 bytes. If the number of bytes is greater than the page size (4 bytes), then the keyboard wedge will continue to read the following pages until the correct number of bytes have been read. However, the keyboard wedge will not produce any output if an attempt to read beyond the end of the card memory is made.

## A.12 MIFARE DESFire and MIFARE DESFire EV1 Custom Fields

### A.12.1 App ID

The App ID is the ID of the MIFARE DESFire application to read. This is an integer in the range 1 to 0xFFFFFFFF (zero is reserved for the PICC master application). ISO application identifiers are not supported.

**Note:** MIFARE DESFire commands and responses encode integer values in little-endian format (LSB first). Therefore this must be taken into account when converting the AID from raw bytes to an integer.

### A.12.2 File Num

File Num is the ID of the file on the application to read. This is an integer in the range 0 to 0x1F. ISO file names are not supported.

### A.12.3 Offset

This is treated the same as the offset parameter used by the MIFARE DESFire read commands. For reading standard data or backup data files, it specifies the position within the file, in bytes, from which the read will start. For value files, the value should be less than or equal to four bytes. For record files, it specifies the first record to start reading from.

### A.12.4 Length

This is treated the same as the length parameter used by the MIFARE DESFire read commands. For reading standard data or backup data files, it specifies the number of bytes to read in bytes. For value files, it must be in the range 0 to 4 bytes inclusive. For record files it specifies the number of records to read.

### A.12.5 Card Key

This specifies the key number on the MIFARE DESFire card to use for authentication. Valid values for the card key are in the range zero to thirteen.

### A.12.6 Rdr Key

This specifies the reader key slot to use for authentication. Values in the range 0-255 will be accepted. However, the reader will normally expect MIFARE DESFire keys to be loaded to slots 240-255.

### A.12.7 Auth

This should be enabled if the file requires authentication to be read, for example if the access conditions for the file do not specify free access.

### A.12.8 File Type

This specifies the type of MIFARE DESFire file being read. The available types are standard data, backup data, value, linear record and cyclical record. Refer to the MIFARE DESFire datasheet for further information. This file type will be determined based on the command used to create the file when the MIFARE DESFire card was provisioned.

### A.12.9 File Comms

This determines the communication type to use when reading the card. The available options are none (no encryption or authentication), MACed (message authenticated, but no encryption) and Encrypt (message signed and encrypted). If the card has not been authenticated, this should be set to none. If authentication has been used, the value should be chosen based on the communication settings used when creating the file to be read.

## A.13 MIFARE DESFire EV1 and MIFARE DESFire EV2 Custom Fields

### A.13.1 Start

This is treated the same as the offset parameter used by the MIFARE DESFire read commands. For reading standard data or backup data files it specifies the position within the file, in bytes, from which the read will start. For value files, the value should be less than or equal to four bytes. For record files, it specifies the first record to start reading from.

### A.13.2 Len

This is treated the same as the length parameter used by the MIFARE DESFire read commands. For reading standard data or backup data files, it specifies the number of bytes to read in bytes. For value files, it must be in the range 0 to 4 bytes inclusive. For record files, it specifies the number of records to read.

### A.13.3 Encryption

The encryption option specifies the algorithm to use for encryption during authentication, message signing (MACing) and message encryption.

- The option DES/3DES should be used for both two key triple DES and single key triple DES.
- For three key triple DES, the option 3K3DES should be chosen.
- AES encryption is also supported via the AES option.

### A.13.4 AV1 Diversify (MIFARE DESFire EV1 only)

Specifies the encryption algorithm used when diversifying the MIFARE DESFire key. This will normally match the algorithm used for authentication. This encryption algorithm will then be used to diversify the master key with the same algorithm as used by the MIFARE AV1 SAM, with a diversification input made up of the card key number followed by the card UID. Every algorithm might be based on key number (KN) and the card's UID, or on the cascade tag (CT) and the card's UID.

### A.13.5 CT value

The CT value (cascade tag) is the value (combined with card's UID) used for the key diversification algorithm. This parameter works in collaboration with the AV1 Diversity field. Depending on the chosen algorithm, the key number (KN) or cascade tag (CT) is used. By default, the CT value is equal to 0x88. It is possible for you to specify a different value used for diversification, in the CT value field.

## A.14 PIV Specific Fields

### A.14.1 FASC-N

If enabled, this outputs the entire FASC-N value as defined in the PIV specification. The FASC-N is contained within the Card Holder Unique Identifier (CHUID) of the PIV card. For further details of the FASC-N, refer to the document *Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems* from the US Government Smart Card Interagency Advisory Board.

### A.14.2 GUID

If enabled, this outputs the entire Global Unique Identifier (GUID). The GUID is part of the CHUID. Refer to the document, *Technical Implementation Guidance: Smart Card Enabled Physical Access Control Systems* from the US Government Smart Card Interagency Advisory Board for further details.

### A.14.3 75-Bit GSA

This is a special option to output the CHUID data in the special GSA-75 bit format defined in the SIA white paper, *Physical Access Control System (PACS) in a Federal Identity, Credentialing and Access Management (FICAM) Framework*. This format is a cut-down version of the CHUID typically used in the access control industry.

### A.14.4 FASC-N Custom

This provides the option to output individual parts of the FASC-N. The offset value is the number of bits to start reading data from and the length is the number of bits to read. Up to 9 custom fields are provided to allow all of the components of the FASC-N to be output. Each custom can be set independently of the others.

### A.14.5 FASC-N Custom Remove Parity

If this option is set, then parity bits will be removed from the FASC-N data before starting to process the FASC-N custom fields.

### A.14.6 FASC-N Reverse BCN

If enabled, then each individual nibble of the output will have its bit order reversed before outputting the data.

## A.15 CEPAS Custom Fields

### A.15.1 CAN

CAN is a unique, CEPAS-specific value that can be read by the OMNIKEY® 5x27CK. The offset and length options allow you to specify, in bits, individual parts of the CEPAS CAN to be output, in exactly the same way as with custom PACS output for HID cards.

# Appendix B

## Extended ASCII character set

The following character set is used for extended ASCII character codes from 128 to 254.

ASCII (dec)	Unicode	Char	ASCII (dec)	Unicode	Char	ASCII (dec)	Unicode	Char	ASCII (dec)	Unicode	Char
128	00C7	Ç	160	00E1	á	192	2514	Ł	224	03B1	α
129	00FC	ü	161	00ED	í	193	2534	ł	225	00DF	β
130	00E9	é	162	00F3	ó	194	252C	Ṭ	226	0393	Γ
131	00E2	â	163	00FA	ú	195	251C	Ṫ	227	03C0	π
132	00E4	ä	164	00F1	ñ	196	2500	–	228	03A3	Σ
133	00E0	à	165	00D1	Ñ	197	253C	†	229	03C3	σ
134	00E5	å	166	00AA	ª	198	255E	‡	230	00B5	μ
135	00E7	ç	167	00BA	º	199	255F	‡	231	03C4	τ
136	00EA	ê	168	00BF	¿	200	255A	ℒ	232	03A6	φ
137	00EB	ë	169	2310	ƒ	201	2554	℔	233	0398	Θ
138	00E8	è	170	00AC	¬	202	2569	℔	234	03A9	Ω
139	00EF	ï	171	00BD	½	203	2566	Ṫ	235	03B4	δ
140	00EE	î	172	00BC	¼	204	2560	Ṫ	236	221E	∞
141	00EC	ì	173	00A1	¡	205	2550	=	237	03C6	φ
142	00C4	Ä	174	00AB	«	206	256C	Ṫ	238	03B5	ε
143	00C5	Å	175	00BB	»	207	2567	±	239	2229	∩
144	00C9	É	176	2591	⋮	208	2568	℔	240	2261	≡
145	00E6	æ	177	2592	⋮	209	2564	Ṫ	241	00B1	±
146	00C6	Æ	178	2593	⋮	210	2565	Ṫ	242	2265	≥
147	00F4	ô	179	2502		211	2559	ℒ	243	2264	≤
148	00F6	ö	180	2524	†	212	2558	ℒ	244	2320	ƒ
149	00F2	ò	181	2561	‡	213	2552	℔	245	2321	↓
150	00FB	û	182	2562	Ṫ	214	2553	Ṫ	246	00F7	÷
151	00F9	ù	183	2556	Ṫ	215	256B	Ṫ	247	2248	≈
152	00FF	ÿ	184	2555	Ṫ	216	256A	‡	248	00B0	°
153	00D6	Ö	185	2563	Ṫ	217	2518	↓	249	2219	•
154	00DC	Ü	186	2551	Ṫ	218	250C	ƒ	250	00B7	·
155	00A2	¢	187	2557	Ṫ	219	2588	■	251	221A	√
156	00A3	£	188	255D	Ṫ	220	2584	■	252	207F	ⁿ
157	00A5	¥	189	255C	Ṫ	221	258C	■	253	00B2	²
158	20A7	ƒ	190	255B	Ṫ	222	2590	■	254	25A0	■
159	0192	f	191	2510	ƒ	223	2580	■			

