

## CardMan<sup>®</sup> 5125 - Registry Settings

---

This manual gives an overview in which way the information of proximity cards is read out by the CardMan 5125 driver and mapped into the ATR.  
The specification which card format is used and shall be decoded, is done with registry entries.

---

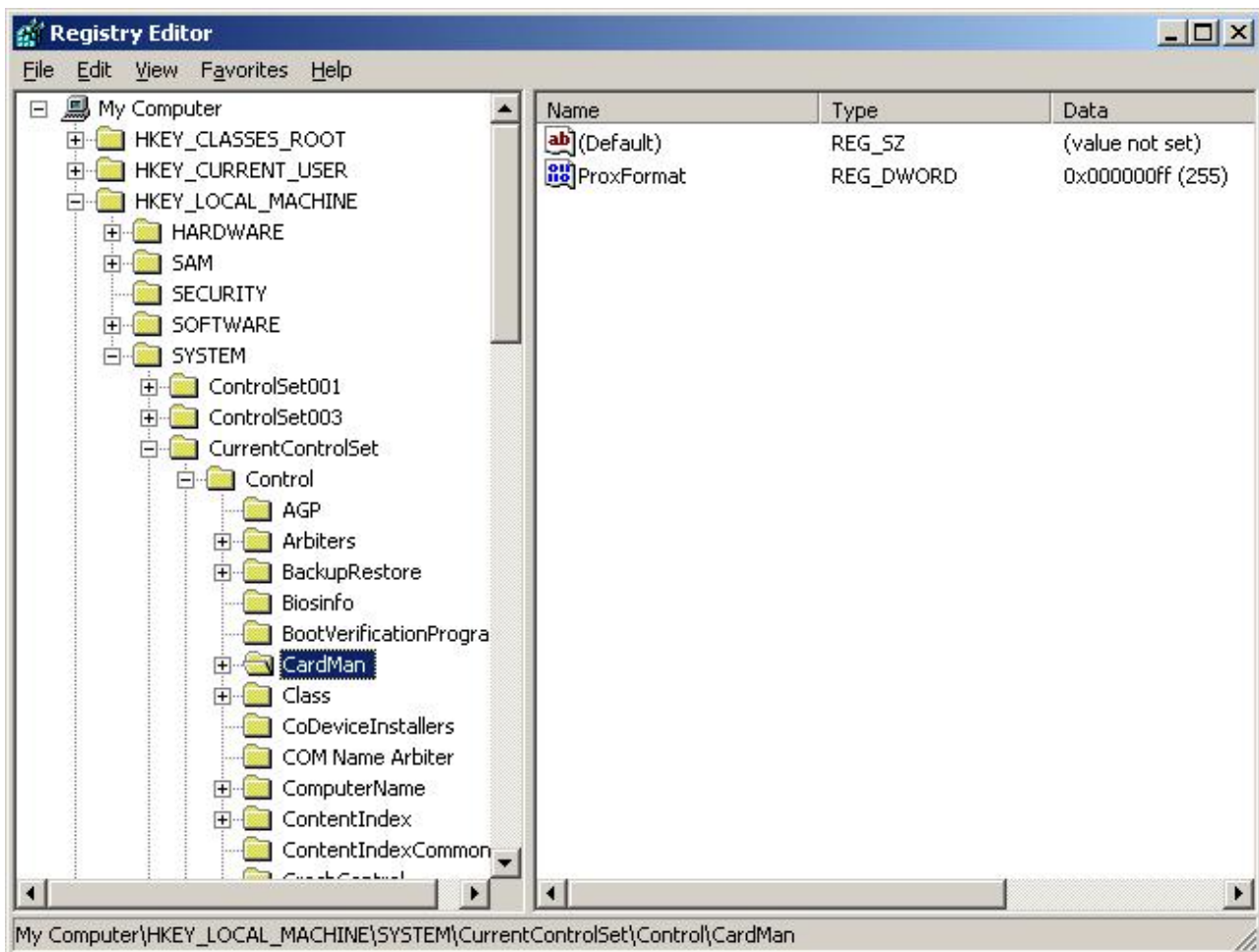


## CardMan<sup>®</sup> 5125 - Registry Settings

### General

The following registry entry specifies the used card format: (see screenshot)

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\CardMan]  
"ProxFormat"=dword:000000ff



To receive further information, please contact OMNIKEY  
E-Mail: [support@omnikey.com](mailto:support@omnikey.com) or visit our homepage [www.omnikey.com](http://www.omnikey.com)

# CardMan<sup>®</sup> 5125 - Registry Settings

The table below shows which entries for the registry key are allowed. If there is a void value, the driver works like “ProxFormat”=dword:00000000 is entered. (= no decoding)

ProxFormat Value Decimal	Card format	Data content
0	Wiegand Raw	-
1	H10301	26 bit (FAC+CN)
2	H10302	37 bit (CN)
4	H10304	37 bit (FAC+CN)
20	H10320	32 bit clock/data card
100	Corp 1000	35 bit (CIC+CN)
254	AUTO	Automatic mode
255	CUSTOMER	Customer defined

FAC ..... Facility Code  
 CN ..... Card Number  
 CIC ..... Customer Identifier Code

Detailed information about the card formats you get from [www.hidcorp.com](http://www.hidcorp.com)

The structure of the decoded ATR depends on the card format and the used registry key. The next table shows in which way the card information is mapped into the ATR depending to the “ProxFormat” value.

FORMAT	CARD		Registry Key (hex format)	Decoded ATR
	FC / CIC	CN		
H10301	AAAA	BBBBBBB	01	3B 06 01 AA AA BB BB BB
H10302	-	BBBBBBBBBBBB	02	3B 07 02 BB BB BB BB BB BB
H10304	AAAAA	BBBBBB	04	3B 07 04 0A AA AA BB BB BB
H10320	-	BBBBBBBBB	14	3B 05 14 BB BB BB BB
Corp 1k	AAAA	BBBBBBBBB	64	3B 07 64 AA AA BB BB BB BB

## CardMan<sup>®</sup> 5125 - Registry Settings

### Automatic Mode

---

If the value of the “ProxFormat” key is set to 254 (0xfe), the detection of the card format and the conversation of the ATR is done automatically by the driver. The function of the automatic mode is restricted because of many different card formats.

Example: The only difference between the two 37bit formats H10302 and H10304 is that H10304 contains a facility code in the ATR and the H10302 not. So it's not possible to difference the two formats on the basic of the ATR.

Because of this, the automatic mode supports and decodes the following formats right:

**H10301**

**H10302**

**H10320**

**Corp 1000**

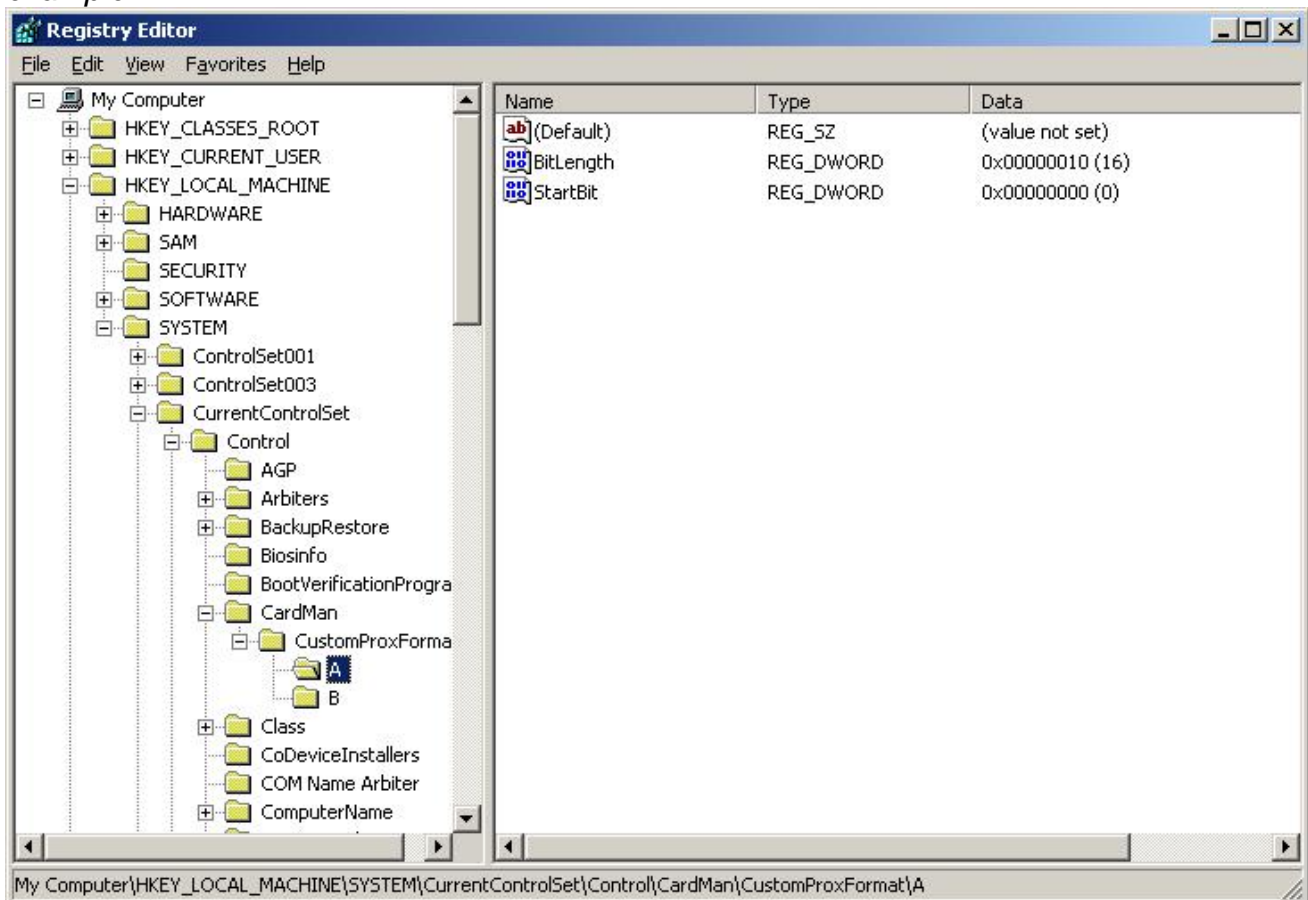
## CardMan<sup>®</sup> 5125 - Registry Settings

### Custom Mode Windows

Because of many different card formats, the driver provides an option to decode the ATR via custom settings. To enable this function the “ProxFormat” key must be set to 255 (0xff). This chapter explains in which way the driver is decoding the ATR by setting additional registry keys.

The bit-data-stream can be splitted into up to 15 several data-fields. Each datafield is labeled with a letter (A, B, ...) and is defined with a StartBit and the BitLength. The StartBit specifies the position in the bit-data-stream, starting with the LSB. So in the case, that the first bit of the data-field shall be the LSB, the value of StartBit must be 0.

#### example



The data-fields are converted into BCD format next and mapped into the ATR in the sequence A B C ....

# CardMan<sup>®</sup> 5125 - Registry Settings

## Example: H10301 format

26bit code (24 information bits)

Card		w/o RegKey	ProxFormat = 255
FC=1	CN=12345	3B 05 00 02 02 60 73	3B 06 01 00 01 01 23 45

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat]

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\A]

"StartBit"=dword:00000011

"BitLength"=dword:00000008

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\B]

"StartBit"=dword:00000001

"BitLength"=dword:00000010

P AAAAAAAAAABBBBBBBBBBBBBBBBBBBP  
 1 0000000100110000001110011

- P Parity Bit
- A Facility Code (FAC)
- B Card Number (CN)

both datafields converted into BCD format:

00000001 -> 00 01 (4 digits, defined by H10301 format)  
 0011000000111001 -> 01 23 45 (6 digits, defined by H10301 format)

# CardMan<sup>®</sup> 5125 - Registry Settings

## Example: H10302 format

37bit code (35 information bits, CN)

Card	w/o RegKey	ProxFormat = 255
CN=1	3B 06 00 00 00 00 00 02	3B 07 02 00 00 00 00 01

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat]

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\A]

"StartBit"=dword:00000001

"BitLength"=dword:00000023

```
PAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA P
000000000000000000000000000000000000010
```

P Parity Bit  
A Card Number (CN)

datafield converted into BCD format:

```
000000000000000000000000000000000001 -> 0 00 00 00 00 01
(11 digits, defined by H10302 format)
```

# CardMan<sup>®</sup> 5125 - Registry Settings

## Example: H10304 format

37bit code (35 information bits, FC+CN)

Card		w/o RegKey	ProxFormat = 255
FC=65535	CN=524287	3B 06 00 0F FF FF FF FF	3B 07 04 06 55 35 52 42 87

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat]

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\A]

"StartBit"=dword:00000014  
 "BitLength"=dword:00000010

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\B]

"StartBit"=dword:00000001  
 "BitLength"=dword:00000013

```
PAAAAAAAAAAAAAAAAA BBBBBBBBBBBBBBBBBBBB P
0 1111111111111111 1111111111111111 1
```

- P Parity Bit
- A Facility Code (FAC)
- B Card Number (CN)

both datafields converted into BCD format:

```
1111111111111111 -> 6 55 35 (5 digits, defined by H10304 format)
1111111111111111 -> 52 42 87 (6 digits, defined by H10304 format)
```

# CardMan<sup>®</sup> 5125 - Registry Settings

## Example: Corp 1000 format

35bit code (32 information bits, FC+CN)

Card		w/o RegKey	ProxFormat = 255
FC=4095	CN=2	3B 06 00 03 FF E0 00 05	3B 07 64 40 95 00 00 00 02

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat]

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\A]

"StartBit"=dword:00000015  
 "BitLength"=dword:0000000C

[HKLM\SYSTEM\CurrentControlSet\Control\CardMan\CustomProxFormat\B]

"StartBit"=dword:00000001  
 "BitLength"=dword:00000014

PPAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBP  
 011111111111110000000000000000000101

- P Parity Bit
- A Facility Code (FAC)
- B Card Number (CN)

both datafields converted into BCD format:

111111111111 -> 40 95  
 (4 digits, defined by Corp1000 format)  
 0000000000000000000010 -> 00 00 00 02  
 (8 digits, defined by Corp1000 format)

## CardMan<sup>®</sup> 5125 - Registry Settings

### Custom Mode Linux & Mac OS X

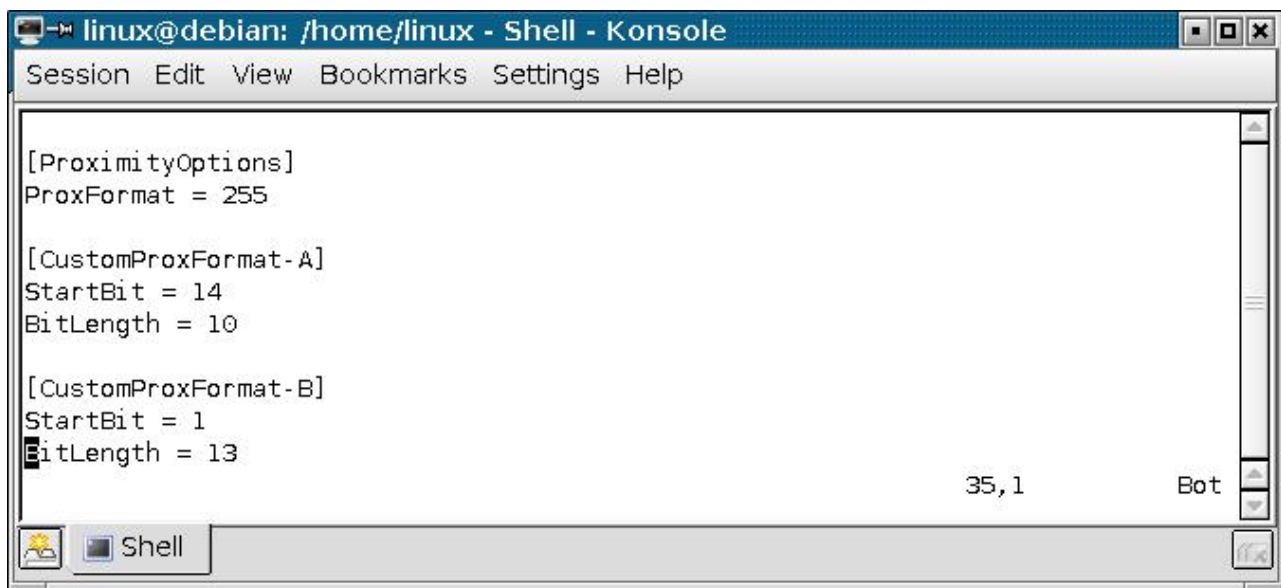
Because Linux and Mac OS X based operating systems do not have a registry the prox specific settings have to be done in an extra file.

This file is named cmrfid.ini and it is copied to the /etc/ directory (the directory is the same on Mac OS X and Linux platform) during driver installation. Editing this file can be done with any text editor, notice that you need root permissions to be able to edit this file.

By default the ProxFormat mode is set to automatic:

```
[ProximityOptions]
ProxFormat = 254 (0xfe)
```

To enable the custom mode function on Linux the "ProxFormat" key must be set to 255 (0xff). Additionally you will need to add entries regarding to the needs of the used card.

A screenshot of a Linux terminal window titled 'linux@debian: /home/linux - Shell - Konsole'. The window shows the contents of a file named cmrfid.ini. The file contains three sections: [ProximityOptions] with ProxFormat = 255, [CustomProxFormat-A] with StartBit = 14 and BitLength = 10, and [CustomProxFormat-B] with StartBit = 1 and BitLength = 13. The terminal also shows a status bar at the bottom with '35,1' and 'Bot'.

In this screenshot a H10304 format card was added.

#### In detail:

First the ProxFormat value in the [ProximityOptions] section has to be changed to 255 (custom mode):

## CardMan<sup>®</sup> 5125 - Registry Settings

```
[ProximityOptions]
; ProxFormat = 254 (= automatic mode)
ProxFormat = 255
```

Then you will need to add the entries for the format options. On Windows based operating systems you would create registry keys as mentioned in the sections above. On Linux you need to create additional format sections for the used card (H10301 in this example):

```
[CustomProxFormat-A]
StartBit = 11
BitLength = 8
```

```
[CustomProxFormat-B]
StartBit = 1
BitLength = 10
```

For any other card please refer to the different examples above, the settings are the same as for Windows, the only difference is that the settings have to be done in the cmrfid.ini file.