



pcProx[®] Config

Configuration Utility
User Manual

Thank You!

Congratulations on the purchase of your pcProx, AIR ID and/or Wiegand device(s). RF IDEas hopes you enjoy using the readers as much as we enjoyed creating and developing them. Configuration is easy, so you will be able to quickly take advantage of a more secure environment in your business, school, or organization.

Please call our Sales department if you have any questions or are interested in our OEM and Independent Developer's programs.

We look forward to your comments and suggestions for our product line! Please go to www.RFIDEas.com and follow the **Support** ⇒ **Learning Center** link for more details about our product line.

We are always discovering new applications for our product line(s). There are several software developers licensing our technology so the solution you are looking for may already be developed.

Thank you,
The RF IDEas Staff

Need Assistance?

Ph: 847.870.1723

Fx: 847.483.1129

E: Sales@RFIDEas.com

TechSupport@RFIDEas.com

END-USER LICENSE AGREEMENT

LICENSE AGREEMENT

End-User License Agreement for RF IDEas™ SOFTWARE and HARDWARE - RF IDEas' pcProx®, AIR ID®, Proximity Activated Readers, Software Developer's Kit, and Proximity Reader DLLs, and Protocol(s).

IMPORTANT-READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and the manufacturer RF IDEas ("Manufacturer") with which you acquired the RF IDEas software and hardware product(s) identified above ("PRODUCT"). The PRODUCT includes the RF IDEas reader, computer software, the associated media, any printed materials, and any "on line" or electronic documentation. By installing, copying or otherwise using the PRODUCT, you agree to be bound by the terms of this EULA. The SOFTWARE PORTION OF THE PRODUCT includes the computer software, the associated media, any printed materials, and any "on line" or electronic documentation. By installing, copying or otherwise using the PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, RF IDEas is unwilling to license the PRODUCT to you. In such event, you may not use or copy the SOFTWARE PORTION OF THE PRODUCT, and you should promptly contact the vendor you obtained this PRODUCT from for instructions on return of the unused product(s) for a refund.

The products described in this publication are intended for consumer applications. RF IDEas assumes no liability for the performance of product. RF IDEas products are not suitable for use in life-support applications, biological hazard applications, nuclear control applications, or radioactive areas. None of these products or components, software or hardware, are intended for applications that provide life support or any critical function necessary for the support of protection of life, property or business interests. The user assumes responsibility for the use of any of these products in any such application. RF IDEas shall not be liable for losses due to failure of any of these products, or components of these products, beyond the RF IDEas commercial warranty, limited to the original purchase price.

SOFTWARE PRODUCT LICENSE The PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PORTION OF THE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This EULA grants you the following rights: *Software. You may install and use one copy of the SOFTWARE PORTION OF THE PRODUCT on the COMPUTER. *Network Services. If the SOFTWARE PORTION OF THE PRODUCT includes functionality that enables the COMPUTER to act as a network server, any number of computers or workstations may access or otherwise utilize the basic network services of that server. The basic network services are more fully described in the printed materials accompanying the SOFTWARE PORTION OF THE PRODUCT. *Storage/Network Use. You may also store or install a copy of the computer SOFTWARE PORTION OF THE PRODUCT on the COMPUTER to allow your other computers to use the SOFTWARE PORTION OF THE PRODUCT over an internal network, and distribute the SOFTWARE PORTION OF THE PRODUCT to your other computers over an internal network.

1.1 General License Grant RF IDEas grants to an individual, a personal, nonexclusive license to make and use copies of the SOFTWARE PRODUCT for the sole purposes of designing, developing, and testing your software product(s) that are designed to operate in conjunction with any RF IDEas designed proximity reader product. You may install copies of the SOFTWARE PRODUCT on an unlimited number of computers provided that you are the only individual using the SOFTWARE PRODUCT. If you are an entity, RF IDEas grants the right to designate one individual within your organization to have the sole right to use the SOFTWARE PRODUCT in the manner provided above.

1.2 Documentation. This EULA grants an individual, a personal, nonexclusive license to make and use an unlimited number of copies of any documentation, provided that such copies shall be used only for personal purposes and are not to be republished or distributed (either in hard copy or electronic form) beyond the user's premises and with the following exception: you may use documentation identified in the SOFTWARE PRODUCT as the file format specification for RF IDEas' proximity readers solely in connection with your development of software product(s) or an integrated work or product suite whose components include one or more general purpose software products.

1.3 Storage/Network Use. You may also store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT on computers used by a licensed end user in accordance with Section 1.1. A single license for the SOFTWARE PRODUCT may not be shared or used concurrently by other end users.

1.4 Sample Code. RF IDEas grants you the right to use and modify the source code version of those portions of the SOFTWARE PRODUCT identified as "Samples in the SOFTWARE PRODUCT ("Sample Code") for the sole purposes to design, develop, and test your software product(s), and to reproduce and distribute the Sample Code, along with any modifications thereof, only in object code form.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

*Limitations on Reverse Engineering, Decompilation and Disassembly. You may not reverse engineer, decompile, or disassemble the PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation

*You may not reproduce or otherwise emulate, in whole or in part, any form the protocol(s) defined within this PRODUCT for use without a RF IDEas PRODUCT Redistributable Code. If you are authorized and choose to redistribute Sample Code ("Redistributables") as described in Section 1.4, you agree to: (a) distribute the Redistributables in object code only in conjunction with and as a part of a software application product developed by you using the PRODUCT accompanying this EULA that adds significant and primary functionality to the SOFTWARE PRODUCT ("Licensed Product"); (b) not use RF IDEas' name, logo, or trademarks to market the Licensed Product; (c) include a valid copyright notice on the Licensed Product; (d) indemnify, hold harmless, and defend RF IDEas from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of the Licensed Product; (e) otherwise comply with the terms of this EULA; and (g) agree that RF IDEas reserves all rights not expressly granted. You also agree not to permit further distribution of the Redistributables by your end users except: (1) you may permit further redistribution of the Redistributables by your distributors to your end-user customers if your distributors only distribute the Redistributables in conjunction with, and as part of, the Licensed Product and you and your distributors comply with all other terms of this EULA; and (2) in the manner described in Section 1.4.

*Separation of Components. The PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer.

*Single COMPUTER. The PRODUCT is licensed with the COMPUTER as a single integrated product. The PRODUCT may only be used with the COMPUTER.

*Rental. You may not rent or lease the PRODUCT without permission from RF IDEas

*Software Transfer. You may permanently transfer all of your rights under this EULA only as part of a sale or transfer of the COMPUTER, provided you retain no copies, you transfer all of the PRODUCT (including all component parts, the media and printed materials, any

*Separation of Components. The PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer.

*Single COMPUTER. The PRODUCT is licensed with the COMPUTER as a single integrated product. The PRODUCT may only be used with the COMPUTER. *Rental. You may not rent or lease the PRODUCT without permission from RF IDEas.

upgrades, this EULA and, if applicable, the Certificate(s) of Authenticity), AND the recipient agrees to the terms of this EULA. If the PRODUCT is an upgrade, any transfer must include all prior versions of the PRODUCT.

*Termination. Without prejudice to any other rights, RF IDEas may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PORTION OF THE PRODUCT and all of its component parts.

3. UPGRADES. If the SOFTWARE PORTION OF THE PRODUCT is an upgrade from another product, whether from RF IDEas or another supplier, you may use or transfer the PRODUCT only in conjunction with that upgraded product, unless you destroy the upgraded product. If the SOFTWARE PORTION OF THE PRODUCT is an upgrade of a RF IDEas product, you now may use that upgraded product only in accordance with this EULA. If the SOFTWARE PORTION OF THE PRODUCT is an upgrade of a component of a package of software programs which you licensed as a single product, the SOFTWARE PORTION OF THE PRODUCT may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

4. OEM COPYRIGHT. All title and copyrights in and to the PRODUCT (including but not limited to images, photographs, animations, video, audio, music, text and "applets," incorporated into the PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PORTION OF THE PRODUCT, are owned by RF IDEas or its suppliers. The PRODUCT and SOFTWARE PORTION OF THE PRODUCT is protected by copyright laws and international treaty provisions. You may not copy the printed materials accompanying the PRODUCT.

5. DUAL-MEDIA SOFTWARE. You may receive the SOFTWARE PORTION OF THE PRODUCT in more than one medium. Regardless of the type or size of medium you receive, you may use only one medium that is appropriate for your single computer. You may not use or install the other medium on another computer. You may not loan, rent, lease, or otherwise transfer the other medium to another user, except as part of the permanent transfer (as provided above) of the SOFTWARE PORTION OF THE PRODUCT.

6. OEM PRODUCT SUPPORT. Product support for the product is not provided by RF IDEas or its subsidiaries. For product support, please refer to the OEM supplies support number provided in the documentation. Should you have any questions concerning the EULA, or if you desire to contact OEM for any other reason, please refer to the address provided in the documentation provided.

FOR THE LIMITED WARRANTIES AND SPECIAL PROVISIONS PERTAINING TO YOUR PARTICULAR JURISDICTION, PLEASE REFER TO YOUR WARRANTY BOOKLET INCLUDED WITH THIS PACKAGE OR PROVIDED WITH THE SOFTWARE PRODUCT PRINTED MATERIALS.

Limited Warranty: RF IDEas warrants to the original buyer of this product, that the hardware and related disk(s) are free of defects in material and workmanship for a period of one year from date of purchase from RF IDEas or from an authorized RF IDEas dealer. Should the RF IDEas products fail to be in good working order at any time during the one-year period, RF IDEas will, at its option, repair or replace the product at no additional charge, provided that the product has not been abused, misused, repaired or modified. This warranty shall be limited to repair or replacement and in no event shall RF IDEas be liable for any loss of profit or any commercial or other damages, including but not limited to special, incidental, consequential or other similar claims.

No dealer, distributor, company, or person has been authorized to change or add to the terms of this agreement, and RF IDEas will not be bound by any representation to the contrary. RF IDEas SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS OF PURPOSE. Since some states do not allow such exclusion of limitation of incidental or consequential damages for consumer products, check the statute of the state in which your business resides. This warranty gives you the specific legal rights in addition to any rights that you have under the laws of the state in which your business resides or operates.


Returns: RF IDEas products which require Limited Warranty service during the warranty period shall be delivered to the nearest authorized dealer or sent directly to RF IDEas at the address below with proof of purchase and a Return Materials Authorization (RMA) Number provided by RF IDEas Technical Support Dept. Replacement parts or complete boards become the property of RF IDEas. If the returned board or unit is sent by mail, the purchaser agrees to pre-pay the shipping charges and insure the board or unit or assume the risk of loss or damage which may occur in transit. The purchaser is expected to employ a container equivalent to the original packaging.

Copyright: Copyright by RF IDEas 2011. All rights reserved. Reproduction or distribution of this document in whole or in part or in any form is prohibited without express written permission from RF IDEas.

Trademarks: All RF IDEas products are trademarks of RF IDEas. All other product names or names are trademarks or registered trademarks of their respective holders.

Disclaimer: This Reference Guide is printed in the U.S.A. Any resemblance mentioned in the Reference Guide to persons living or dead, or to actual corporations or products is purely coincidental. RF IDEas believes that the information contained in this manual is correct.

However, RF IDEas does not assume any responsibility for the accuracy of the content of this User Manual, nor for any patent infringements or other rights of third parties. RF IDEas reserves the right to make any modifications in either product or the manual without giving prior written notification.

 **CAUTION:** Pursuant to Part 15.21 of the FCC Rules, any changes or modifications to this product not expressly approved by RF IDEas might cause harmful interference and void the FCC authorization to operate this product.

FCC Compliance Statement

FCC ID: M9MPCPROXHUSB100 (HID USB model)	FCC ID: M9MPCPROXH100 (HID RS-232 model)
FCC ID: M9MPCPROXM101 (Indala model)	FCC ID: M9MBUPCPROXA100 (AWID)
FCC ID: M9MRDR6X8X (Kantech, Indala, Casi-Rusco)	FCC ID: M9MPCPROXP100 (Pyramid)
FCC ID: M9MPCPROXC101 (Casi-Rusco model)	FCC ID: M9MRDR7P71 (FIPS 201 13.56MHz)
FCC ID: M9MRFID1356I100 (MIFARE/iCLASS models)	FCC ID: M9MRDR7L81 (Legic 13.56MHz)
FCC ID: M9MRDR7081 (iCLASS Module based)	FCC ID: M9MRDR7580 (iCLASS MIFARE & Other 13.56Mhz)
FCC ID: M9MRDR7581 (iCLASS MIFARE & Other 13.56MHz)	FCC ID: M9MRDR7081AKF (iCLASS MIFARE & Other 13.56MHz)
FCC ID: M9MRDR7081AKE (iCLASS MIFARE & Other 13.56MHz)	FCC ID: M9MRDR75DX (iCLASS MIFARE & Other 13.56MHz)
FCC ID: M9MRDR8XX8U (Plus combo model)	

*Changes to this reader system not expressly approved by RF IDEas will void the User's authority to operate the equipment.

Note: This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. This product complies with FCC OET Bulletin 65 radiation exposure limits set forth for an uncontrolled environment.

The reader may not recognize value cards in the presence of high RF fields. If the current reading is erratic, the user shall take the following step: Move the equipment from any known transmitters nearby. For more information contact Tech Support at 866.439.4884.

Contents

2	Chapter 1: The Basics	47	Chapter 4: ASCII Command Protocol
2	Thank You!	47	ASCII Command Overview
6	Wireless Identification Overview	48	Connect Serial Communications
7	ID Card Reader System	49	Command Structure
7	pcProx Output Formats	52	Help Command
8	pcProx Features	54	Variable Command
8	pcProx Functions	58	ACP Error Codes
8	pcProx Connectors		
10	USB Readers & Wiegand Converters	59	Chapter 5: Tips and Troubleshooting
10	RS-232 Readers & Converters	59	Troubleshooting
10	System Requirements	60	Precautions
10	Card Compatibility	61	Appendix
		62	The pcProx for Password Security
11	Chapter 2: Getting Started		
11	Hardware Installation	63	Index
11	pcProx Software Installation		
		64	Other Products and Accessories
15	Chapter 3: Configuration		
15	pcProx Configuration Utility		
15	Tool Bar		
16	Connect Button		
17	Disconnect Button		
17	Open Button		
18	Save Button		
18	Defaults Button		
19	Flash Button		
19	About Button		
20	File Menu		
20	Connect Menu		
20	Device Menu		
21	Navigation Menu		
21	View Menu		
21	Help Menu		
22	pcProx + Section		
22	Connect Tab		
26	Data Format Tab		
29	Delimiters Tab		
31	Timing Tab		
33	SDK Tab		
37	CHUID Tab		
46	FIPS 201 Card Configuration		

Wireless Identification Overview

pcProx® Activated Identification

Employers are more security conscious than ever. More buildings, machines, systems, and applications require identification information to gain access. RF IDEas devices allow the building access cards to be used as a digital identifier through out the workplace.

pcProx applications include:

- Card Enrollment
- PC/LAN Log On
- Cafeteria Purchases/Vending
- Machine Access
- Time/Attendance

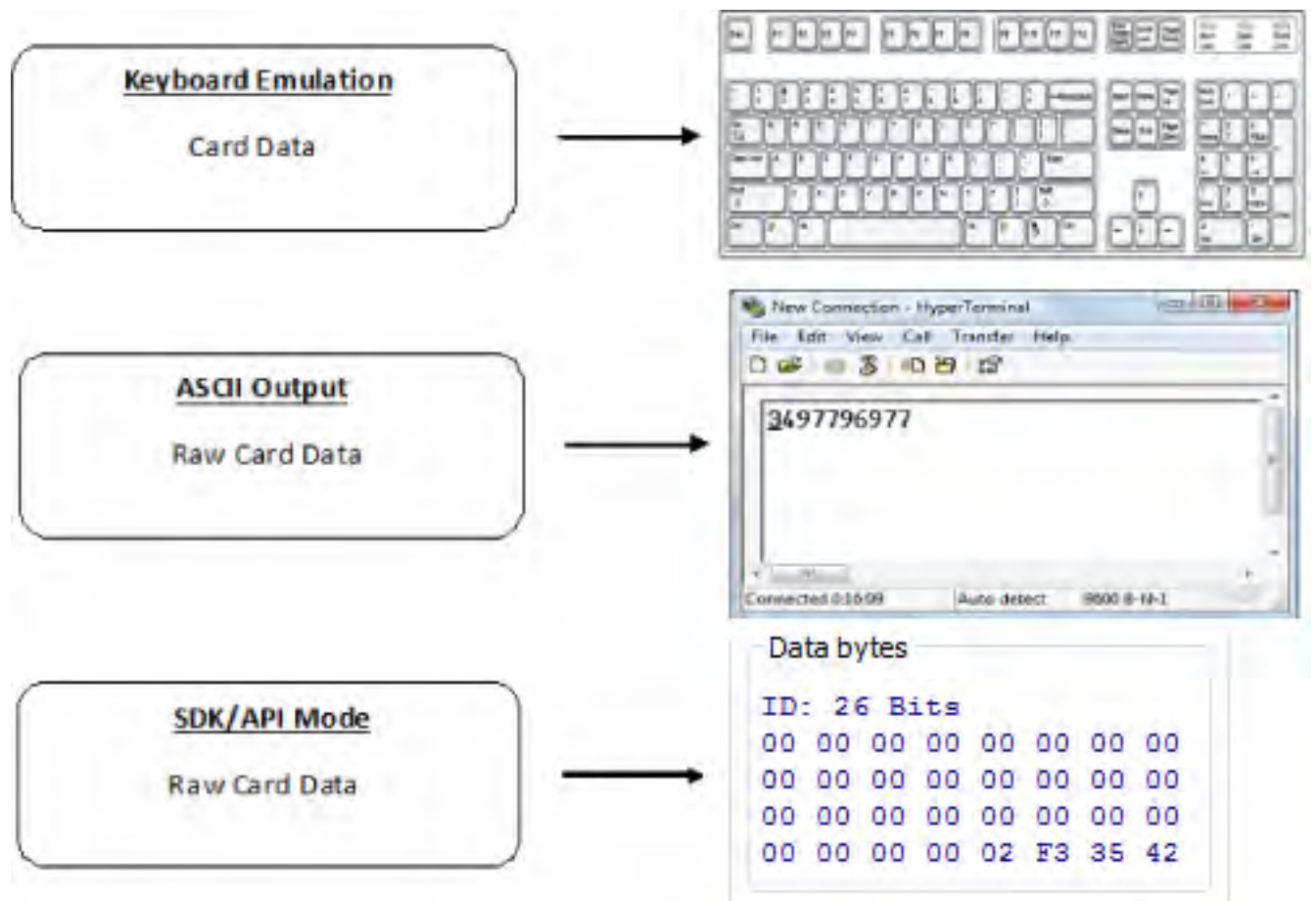
pcProx technology is based on a contactless interface and is not subject to reliability issues. Our pcProx, AIR ID and Wiegand devices are easily configured to increase security and reliability. Companies using proximity and/or contactless technology for building access immediately benefit as their employee identification cards can also be used with the proximity/contactless device for additional authentication applications. Thus, the majority of deployment and enrollment costs are quickly recovered.

The diagram on the following page is a high level overview of how the reader works. The card sends radio signals to the reader and the reader sends these signals back to read the card. The card data is output by the reader in keystrokes or ASCII characters. This card data can be configured to include delimiters to separate the data. A list of features, functions, and connectors follows. This reader can be used as a standalone system or seamlessly integrated with other software applications using the optional Software Developer's Kit (SDK).

ID Card Reader System



Output Formats



Features

- Read all data from proximity/contactless cards
- Read configuration
- Write configurations
- Software Developer's Kit/API compatibility
- Output in decimal or hexadecimal
- User controls number of digits output

Functions

- Software Developer's Kit (SDK) USB
- Software Developer's Kit RS-232
- USB Keyboard
- RS-232
- Serial Virtual COM

Connectors

- USB Keyboard
- USB Virtual COM Port
- RJ45
- PS/2
- DB-9 RS-232
- Power Plug - 2.1 mm
- Power Plug - 2.5 mm
- PS/2
- DB9 - Pin 9 Power

Connectors



CONNECTORS



USB
Keyboard



USB
Virtual COM Port



DB-9
RS-232



Female Power



RJ45



PS/2 Power

OUTPUT



or

RS-232 Data
0.012345679012

Card Types We Support
View the list at www.RFIDeas.com

USB Readers and Wiegand Converters

The USB keystroke reader operates in two primary modes:

- USB keyboard. It reads the card data and sends it as keystrokes as if the user typed the ID data on a keyboard.
- Under the application programmer interface (API) defined in the pcProx SDK. When it reads card data, the active application receives the entire card data.

RS-232 Readers and Converters

The RS-232, Ethernet, or virtual COM port reader operates in two primary modes:

1. ASCII output device. In this mode the user card data is read and sent as a decimal or hexadecimal number in ASCII characters.
2. API defined in the pcProx SDK. The device attaches to a computer serial port. When it reads card data, the active application receives the entire card data.

Once the configuration settings are correctly configured and written to flash memory, the device can immediately be deployed.

	Minimum System Requirements
HARDWARE	Pentium class PC
MEMORY	32 MB RAM
DISK	25 MB hard disk space
I/O	1 available RS-232 or USB Port
Operating System	Any operating system that supports a USB keyboard including Microsoft Windows 2000, XP, Vista, Linux, Macintosh. Can be used for keystroke applications

Note: The software does not perform any data validation checking. The data must be known before it is read to verify its validity.

Manufacturer/Vendor Card Compatibility

Please go to www.RFIDeas.com for specific device part numbers associated to card types.

Hardware Installation

Plug the connector into the workstation's (or available on any peripheral) open RS-232, USB or Ethernet plug.

Place the device next to the monitor, beside the workstation, or where appropriate.

The workstation should detect new hardware for USB connections. Verify the workstation recognizes this connection using the 'Device Manager'.

Verify the correct COM port for RS-232 DB9 connections using the 'Device Manager'.

When the software is installed, it should recognize these connections in order to configure the appropriate device. Once the device is configured and written to its flash memory, these settings will not have to be configured again.

Software Installation

Install the device installation program to the Desktop for quick access. This installer is digitally signed by RF IDEas to authenticate file integrity for your safety .

Open the www.RFIDEas.com website and click **Support** ⇒ **Software and Downloads**.

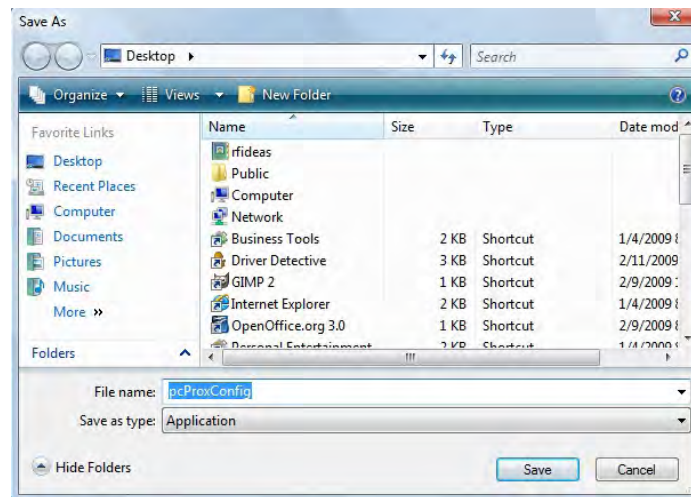
Click **pcProxConfig.exe** to download the installation utility. This file is the pcProx installation program. Use this icon on the desktop to open the installation program to configure the device once the software is installed.

The File Download - Security Warning window displays.

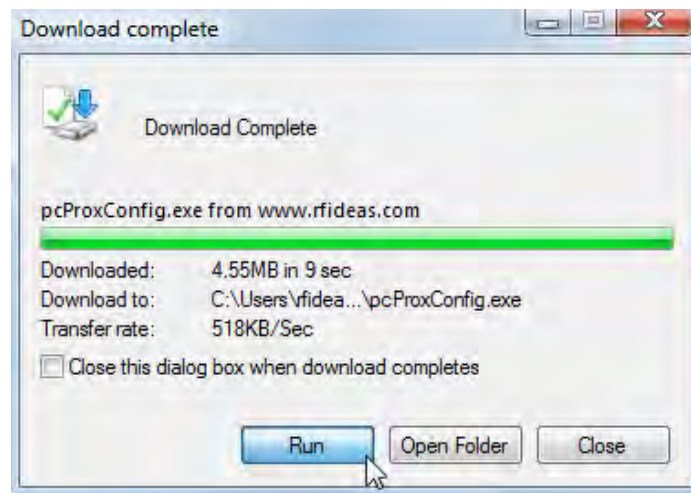


Click **Save** in the File Download - Security Warning window to save the installation program to the Desktop.

Click **Save** in the Save As window to accept the default file name.

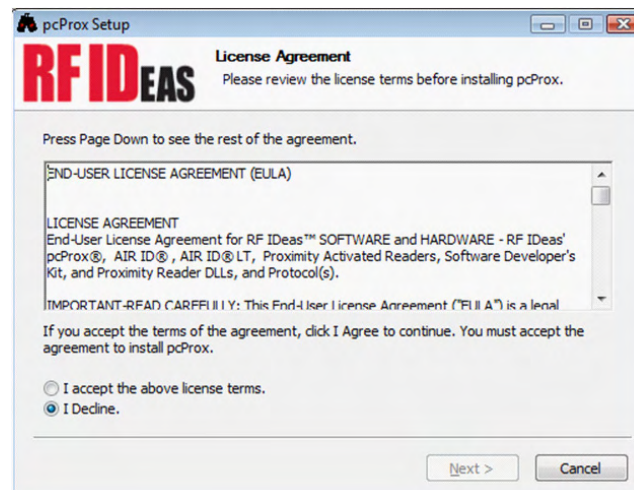


The configuration software downloads to the appropriate location.



Click **Run** in the Download Complete window to install the configuration software.

Follow the prompts to install the configuration software.

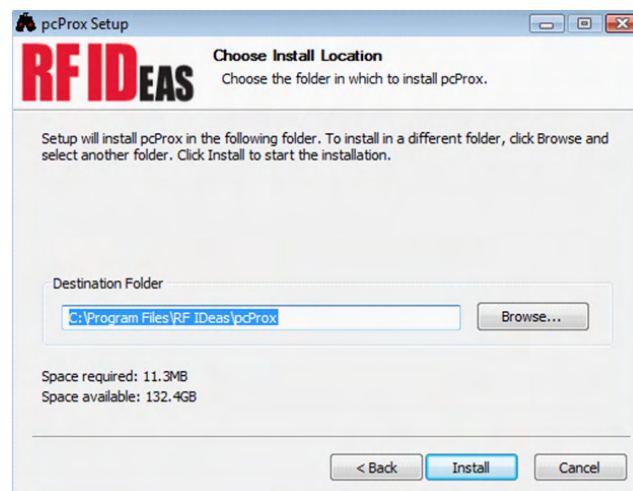


Check pcProxConfig component to install utility and click next.

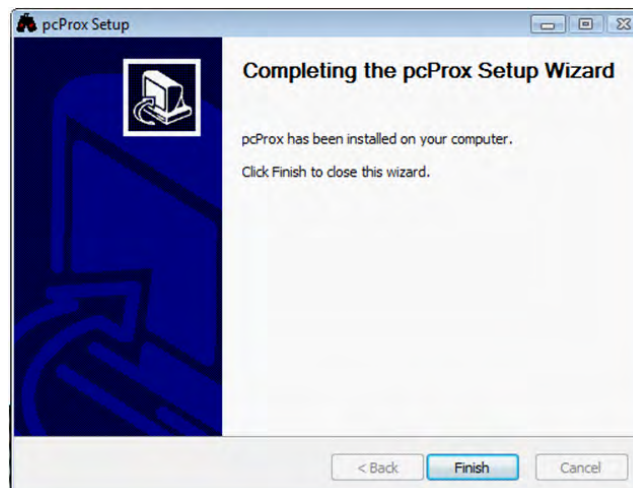


Note: Use **Control Panel** ⇒ **Add/Remove** Programs to successfully remove the setup program if there is a problem with installation or if this is an upgrade. Reboot and then reinstall the program.

Select a destination folder for the utility installation and then click next.



Once the installation is complete, click finish to close the Setup Wizard.



Once the pcProxConfig utility is installed connect a device to the workstation.

Verify the device is connected to the appropriate connector. It is best to configure one device at a time, plug each device in so you know which device is being configured.

Now the device can be configured to output the card data in the appropriate format.

pcProxConfig Utility

The pcProxConfig configuration utility allows for more delimiters to be added with the card data.

Tool Bar



The Tool Bar displays the following commands:

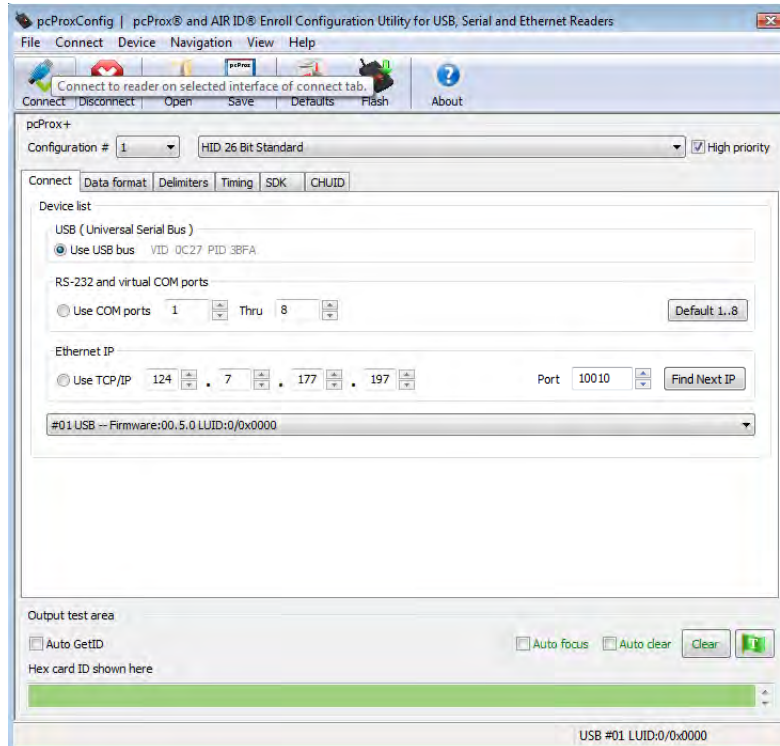
Commands	Click to:
Connect	Connect to reader of selected interface on connect tab.
Disconnect	Disconnect all devices
Open	Opens a specific configuration into the selected device.
Save	Save the configuration as a HWG+ file.
Defaults	Reset the device configuration to the factory default settings.
Flash	Write configuration into device's flash memory. Save's the on screen settings into device's flash memory.
About	Display the application and library version.

If no device is found, the following message displays:



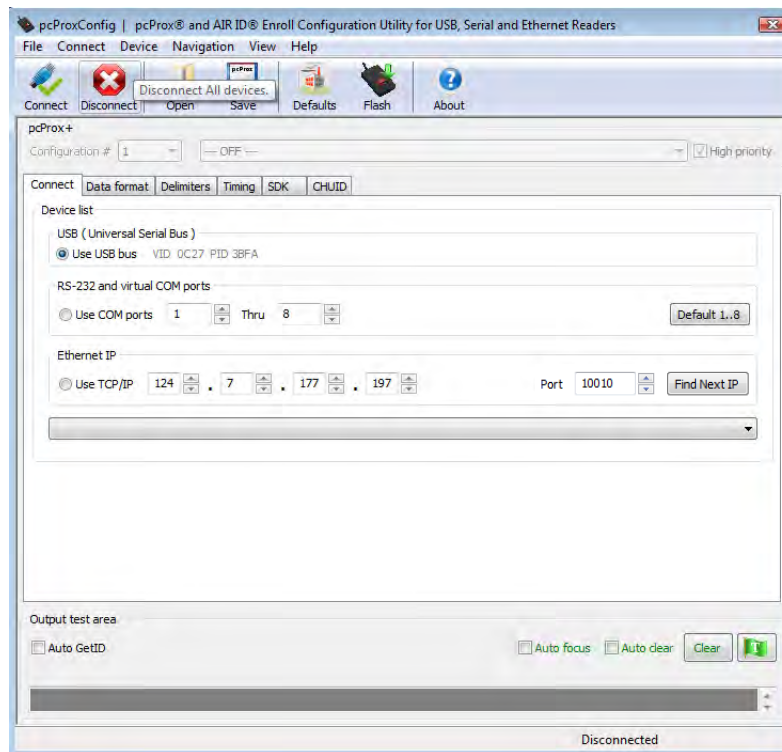
Connect Button

Click **Connect** to search for available connected device.



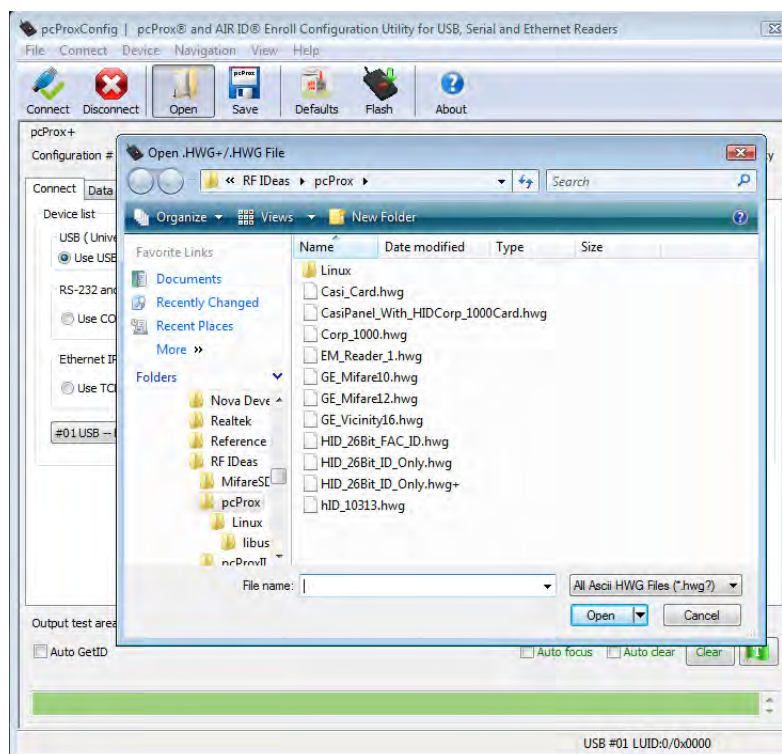
Disconnect Button

Click **Disconnect** to disconnect from available connected device.



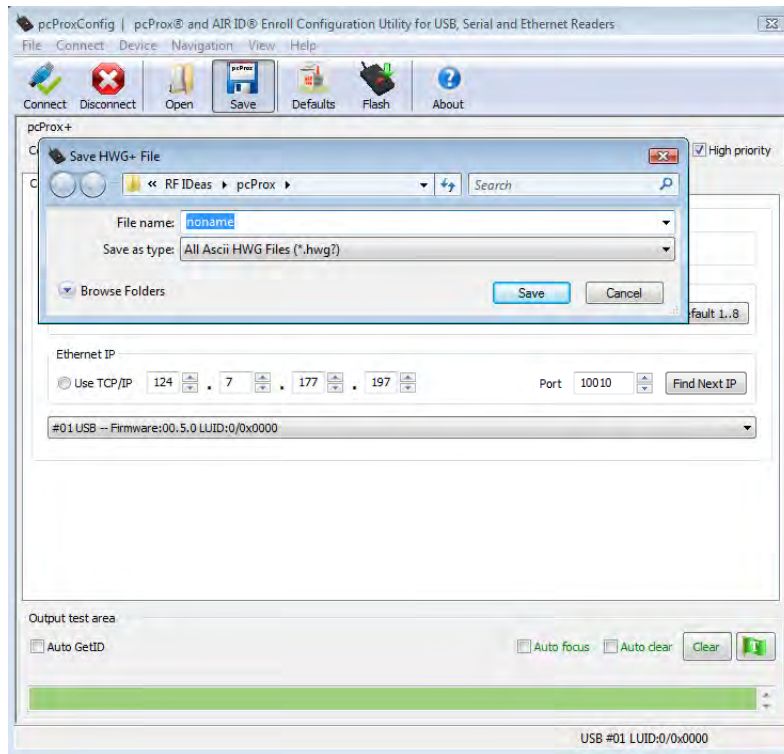
Open Button

Click **Open** to load an ASCII .HWG+ file into the device. The following message displays:



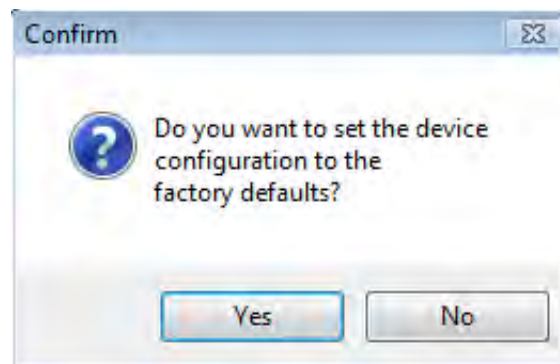
Save Button

The **Save** button will allow the user to save their current device data to an ASCII .HWG+ file for later use.



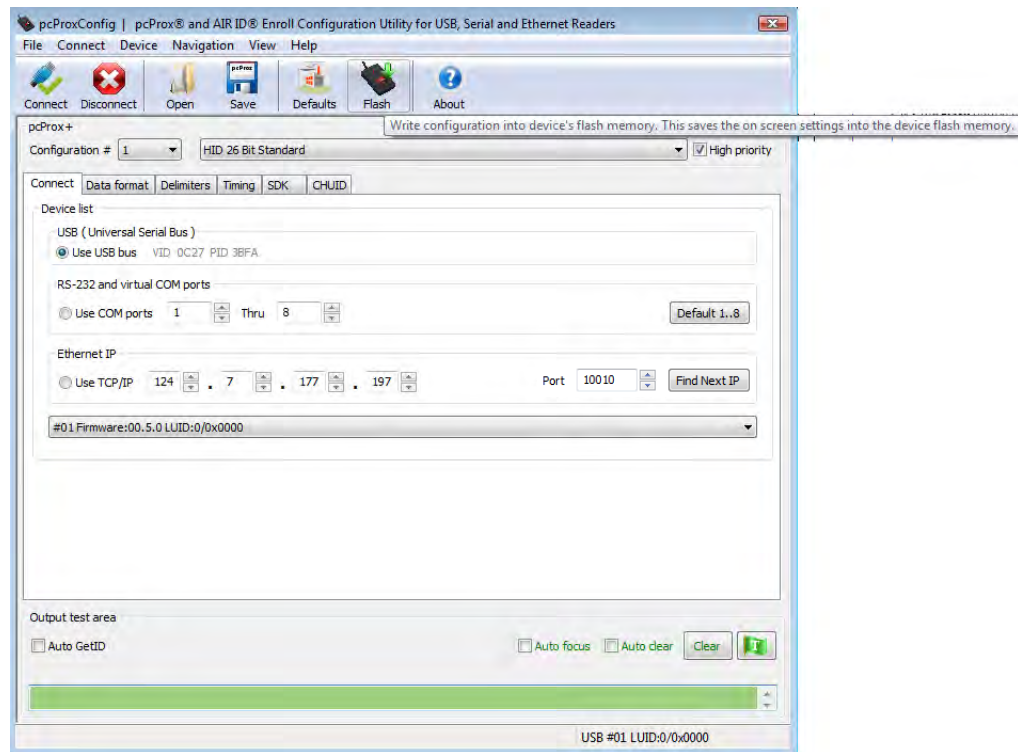
Defaults Button

Click **Defaults** to set the device's flash configuration to factory default settings. The following message will appear for the user to agree or disagree.



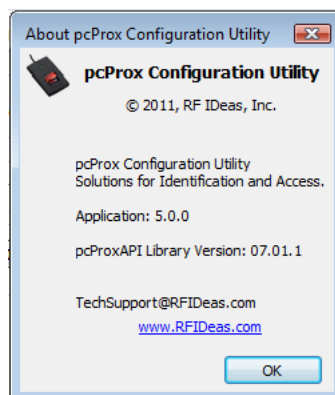
Flash Button

Click the **Flash** button to write the configuration into the device's flash memory.



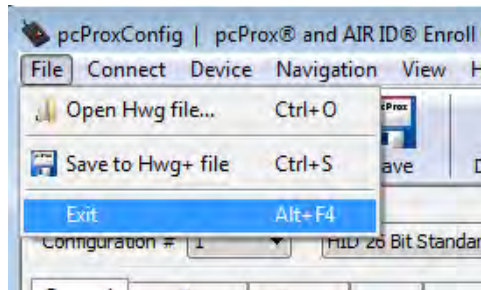
About Button

Click the **About** button to find software information and version number for the configuration utility.



File Menu

The file menu lists the options for Opening .HWG files and Saving to .HWG files. It also includes the Exit command.



Connect Menu

The Connect menu lists the following commands:

Commands	Description
Auto Connect on Startup	Check to connect devices automatically on startup
Connect	Click to connect device
Connect USB	Click to search for USB devices
Connect Serial	Click to search for serial devices
Connect Ethernet TCP/IP	Click to search for Ethernet TCP/IP devices
Disconnect	Click to disconnect device

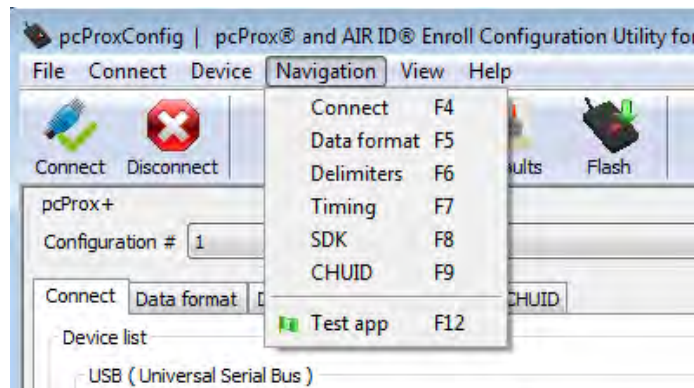
Device Menu

The Device menu lists the following commands:

Commands	Description
Reset Device to Factory Defaults	Click to reset the device to factory defaults
Reset Device to Original Shipping Configuration	Resets device to its original shipping configurations
Save Configuration to Original Shipping Configuration	Click to save a configuration as original shipping configuration.
Read Device Configuration from Flash Memory	Click to read the configuration in the device's flash memory.
Write Configuration to Device Flash Memory	Click to write the current configuration to the device's flash memory.
Clone Current Reader Configuration to other Devices	Click to clone current device configuration to more devices.

Navigation Menu

The Navigation menu lists the same commands as seen on the utility tabs. (An explanation of each can be found in the tab sections of this manual).



View Menu

The View menu allows for the option to change certain visuals on the configuration window.

Commands	Description
Show Tool Tip Balloon Help	Check to allow for pop-up balloons upon hovering over menu option
Show Text Under Toolbar Icons	Check to allow for text to appear under each toolbar icon
Show Pop-up Warning Dialogs	Check to enable pop-up warnings on certain actions (by default, selection is set to Yes)
Show Confirm Dialog Asking Yes/No	Check to have a pop-up for confirmation on certain actions
Beep on Warnings	Check to allow for beeps
Resize Window	Check to allow for resizing of utility window

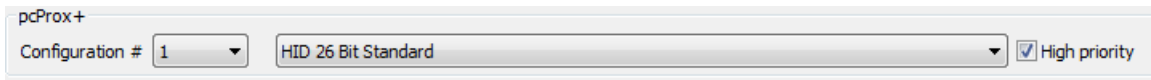
Help Menu

The Help menu allows for the option to resize the configuration window.

Commands	Description
pcProxConfig Manual From Website	Click to be taken to the pcProxConfig manual on the RF IDEas website
www.RFIDEas.com	Click to open the RF IDEas website.
Check Website for Updates	Click to go to the RF IDEas website to check for an update to the configuration utility.
About	Click to display the software name and library versions

pcProx +

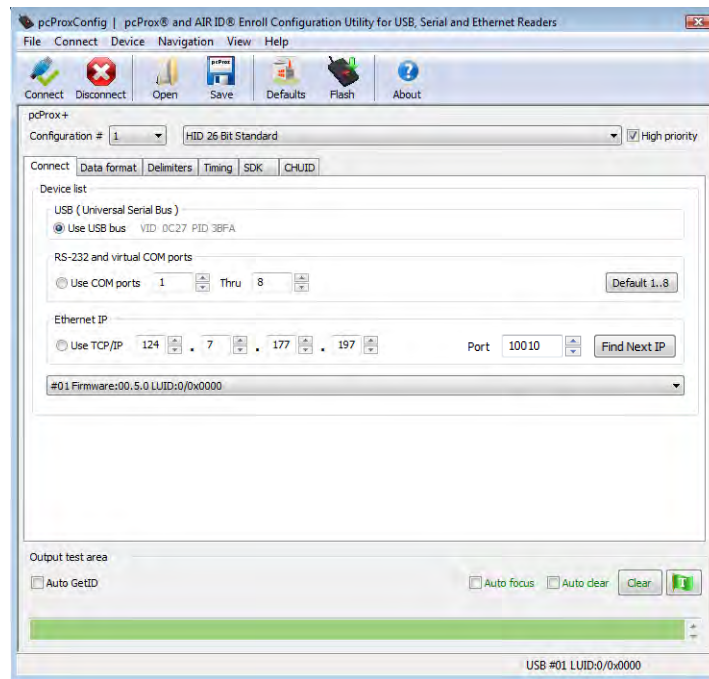
This section allows for users to choose multiple configurations in the **Configuration #** drop down menu. The card type drop down menu allows for a selection of a card type for the configuration. Each configuration (if multiple) can have a separate card type. The **High Priority** checkbox sets the selected configuration as priority above any others.



Note: For example, the High Priority checkbox can set priorities for corp 1000 cards or dual frequency cards. If multiple cards are on the reader the designated High Priority selection will be read first.

Connect Tab

Use this tab to connect to the device.



In the **Connect** tab, the **Device List** allows for the proper interface to be selected to connect the devices.

Select from the following:

USB (Universal Serial Bus)	Scan the USB Bus for readers
RS-232 and Virtual COM Ports	Use serial ports RS-232 and virtual COM ports. Serial devices may slow when scanning a wide port range.
Ethernet IP	Connect to an Ethernet reader at the given IP address, and open a TCP/IP on the given port

USB (Universal Serial Bus)

This selection will scan USB bus for readers.,



USB (Universal Serial Bus)
 Use USB bus VID 0C27 PID 3BFA

RS-232 and Virtual COM Ports

This section scans for RS-232, physical COM port devices, virtual COM port devices, including USB, CDC and PCMCIA devices. It stops after the first USB device is found. The list holds only one serial device.



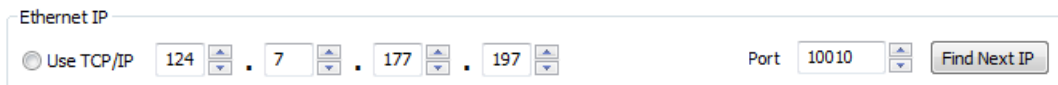
RS-232 and virtual COM ports
 Use COM ports 1 Thru 8 Default 1..8

Once RS-232 selection has been made, the lower and upper limits of the COM ports to scan need to be set. The default COM ports are set at 1 thru 8. The **Default 1..8** button to the right sets the COM ports back to 1 thru 8.

Note: Serial devices may slow when scanning a wide port range.

Ethernet IP

Connect to an Ethernet reader at the given IP address and open a TCP/IP on the given port.

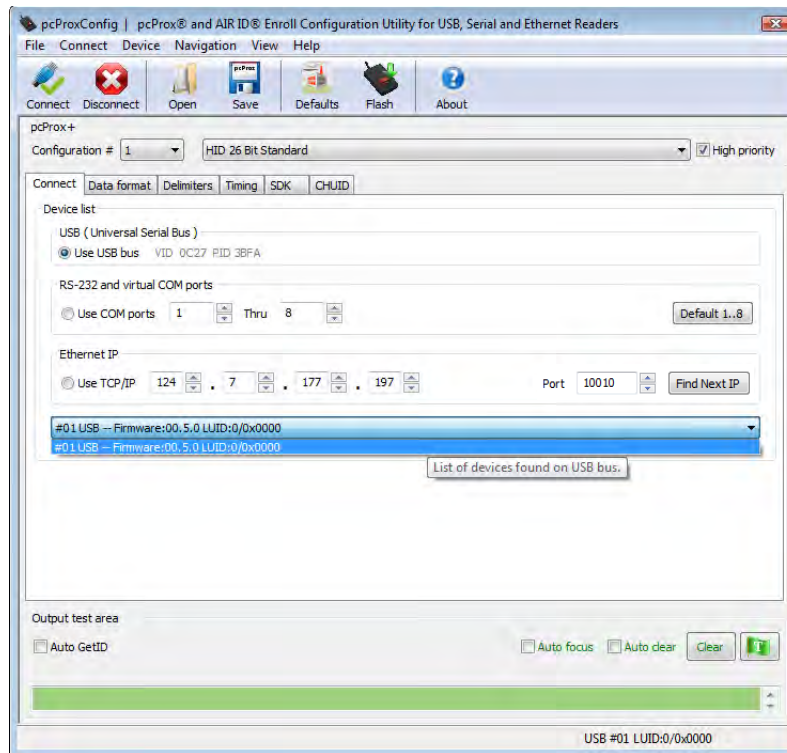


Ethernet IP
 Use TCP/IP 124 . 7 . 177 . 197 Port 10010 Find Next IP

The first, second, third, and fourth byte of the TCP/IP address need to be entered for the interface to connect to the reader. The IP port number will also be required.

Note: Ports below 1024 are for system use only.

The drop-down menu bar will provide the list of devices found on the USB bus.



Output Test Area

This is the test area for the keystrokes entered by the reader. On serial devices this displays the unsolicited serial port data.

The **Auto GetID** box can be checked for the utility to poll the reader for a card ID every 500ms and displays the result to the right of the box.



The **Auto Focus** box keeps the cursor in the test area box to capture the keystrokes output by the device.

Note: When the **Auto Focus** box is checked, it is possible that the selection may conflict with the menus and drop downs, due to the fact that the cursor will attempt to move back into the test area. If this problem arises, simply uncheck the box.

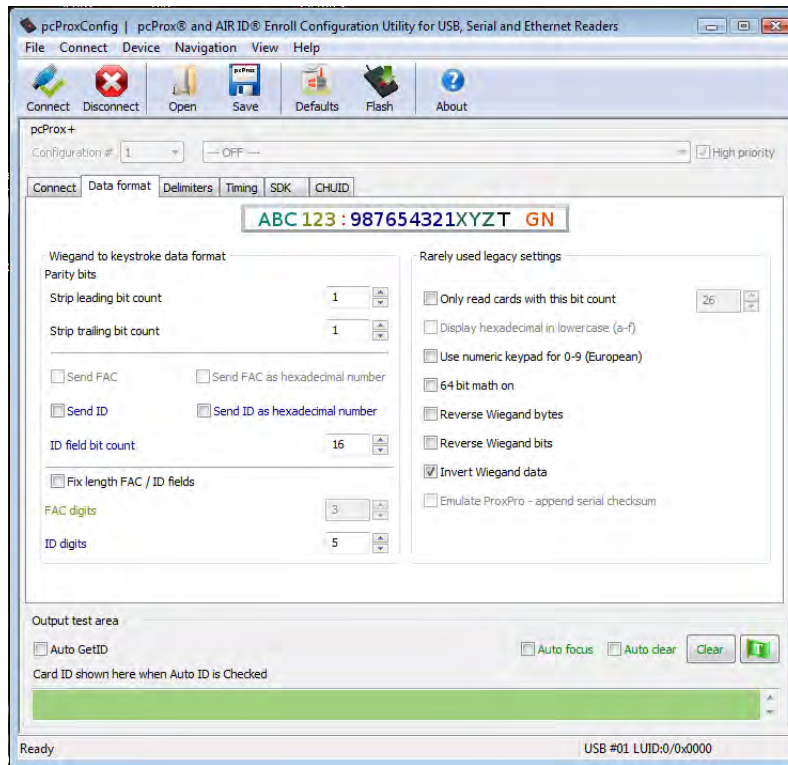
The **Auto Clear** box auto selects all text in the Output Test Area, so that new keystrokes output by the device will replace old text.

The **Clear** button erases all text in the Output Test Area.

The **Test** button (Green Flag) starts the batch file "testarea.bat" or script "testarea" to bring up a users own application to view the readers keystrokes.

Data Format Tab

Use this tab to configure the bits the device reads from the ID token.



Wiegand to Keystroke Data Format

Wiegand to keystroke data format

Parity bits

Strip leading bit count

Strip trailing bit count

Function	Description
Strip leading parity bit count	Set the device to strip leading parity bits from 0 to 15.
Strip trailing parity bit count	Set the device to strip trailing parity bits from 0 to 15.

Send FAC Send FAC as hexadecimal number
 Send ID Send ID as hexadecimal number
 ID field bit count

Function	Description
Send FAC	Check to send the Facility/Site (FAC) code.
Send FAC as hexadecimal number	Check to send this code in hexadecimal. This is set for KANTECH 10 proximity cards.
Send ID	Check to send the ID portion of the card data from the device.
Send ID as hex number	Check to send the ID portion as a hexadecimal number.
ID Field Bit Count	Enter the bit count of ID portion.

Fix length FAC / ID fields
 FAC digits
 ID digits

Function	Description
Fixed Length ID / FAC Fields	Click to set the ID / FAC codes to a fixed length.
ID Digits	Enter the number of zeros to add to the front the ID data to create a specific length.
FAC Digits	Enter the number of zeros to add to the front of the FAC data to create a specific length.

Rarely Used Legacy Settings

Rarely used legacy settings

Only read cards with this bit count 26

Display hexadecimal in lowercase (a-f)

Use numeric keypad for 0-9 (European)

64 bit math on

Reverse Wiegand bytes

Reverse Wiegand bits

Invert Wiegand data

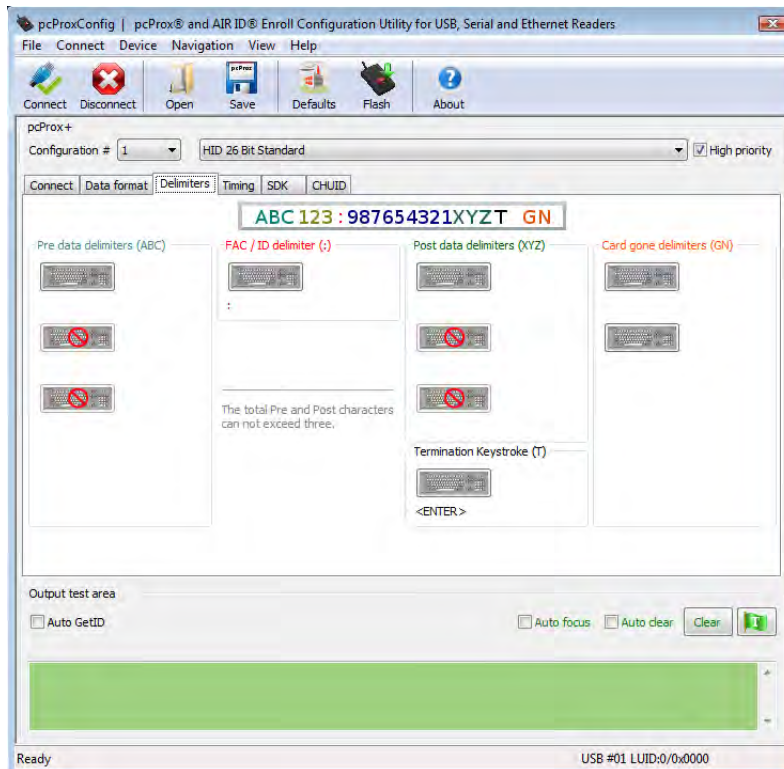
Emulate ProxPro - append serial checksum

Function	Description
Only Read Cards With This Bit Count	No data is sent from the device unless the bit count is matched. The total bits received from the card must match this bit count, parity bits included. If checked and 26 is entered in the field, the device will only respond to 26 bit cards.
Display Hex in Lowercase (a-f)	Check to keystroke out lowercase hex (This option is only available when Send FAC as hex or Send ID as hex is selected)
Use Numeric Keypad for 0-9 (European)	Check to use the European AZERTY keyboard (i.e., keyboard numeric keypad keys). Num Lock must be on.
64 Bit Math On	Check so the device uses a 64 bit binary to decimal conversion to calculate the card number. This is available for firmware version 5.6 and above. In previous versions, the device would only convert 32 bits at a time and concatenate when larger bit length ID numbers were encountered. Check this to display the true representation of the number or if the card is over 32 bits.
Reverse Wiegand Bytes	Check to read the card data in a reverse byte order. Use this feature with MIFARE CSN readers. For 56 bit cards, go to the 'Set Key Stroke Data' tab and set the Bit Count of ID Portion to 56.
Reverse Wiegand Bits	Check to reverse the order of Wiegand bits. This is primarily used for Card Key proximity cards.
Invert Wiegand Data	Check if using a legacy application that requires the Wiegand data to be inverted.
Emulate ProxPro	Emulate serial data format to match HID Corp. Prox Pro reader by sending a 2 byte checksum after the card data.

Delimiters Tab

Use this tab to configure pre and post data delimiters. A delimiter can also be set between the ID and FAC card data.

Click the appropriate keyboard icon to select the pre and post delimiters. Click **Insert**.



Note: Only 3 pre and post delimiters can be configured. If 3 pre-delimiters are set, no post delimiters can be set.

The Scan Code output for the key selected displays above the list of keys.

ABC 123 : 987654321XYZT GN

Pre Data Delimiters (ABC)

Select from 0 to 3 characters to display prior to sending the card data.

FAC/ID Delimiter (:)

Select a character to display between the FAC and ID data.

Post Data Delimiters (XYZ)

Select from 0 to 4 characters to send after the card data is sent. These first three characters are shared between the pre and post string.

Termination Keystroke (T)

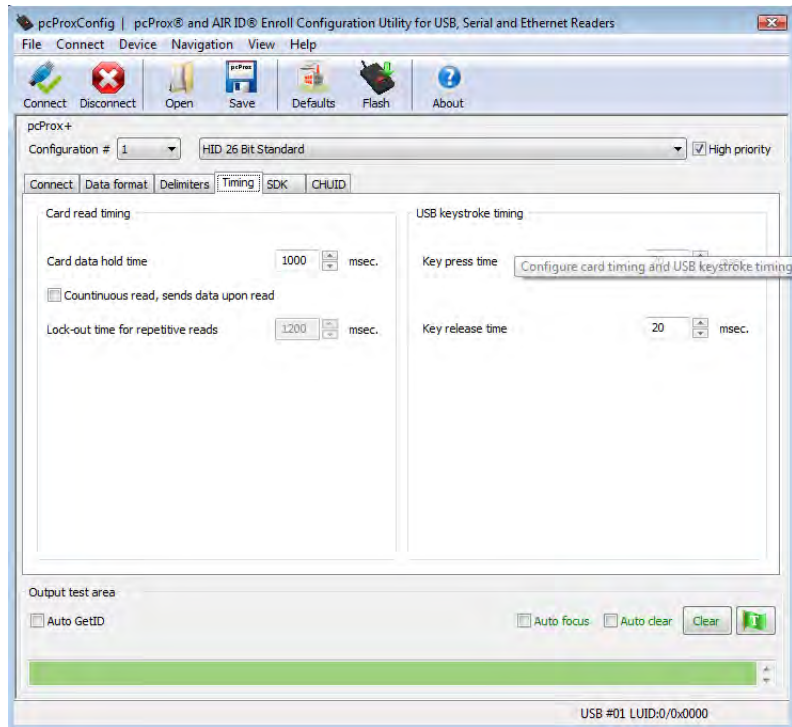
Select a character to display to signify the keystroke termination.

Card Gone Delimiter (GN)

Select two character to display once card data has finished keystroking and leaves the desired field.

Timing Tab

Use this tab to configure the device's card timing and USB keystroke timing.



Card Read Timing

Function	Description
Card Data Hold Time	Enter the time the card data remains valid in the device. The minimum value is 900. This is read in 50 msec increments. The default is 1,000.
Continuous Read, Sends Data Upon Read	If this is checked, the card data is repeatedly sent. Generally this check box is not checked. If left unchecked, the card data is only read once. Otherwise, the card data is continuously sent.
Lock-Out Time For Repetitive Reads	This sets how long the device is locked and will not accept the data of the next card. This is read in 50 msec increments. The minimum value is 0. The maximum is 12,500.

USB Keyboard Timing

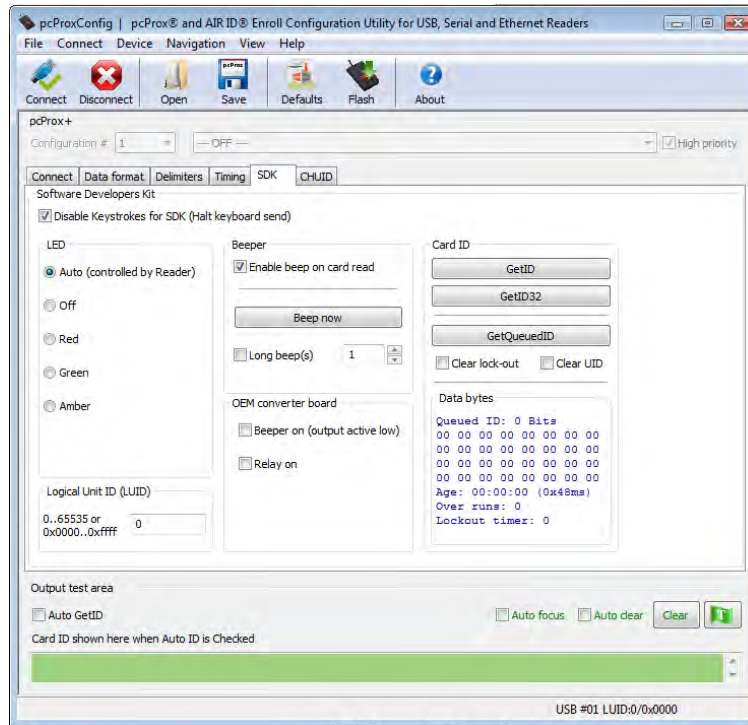
USB keyboard emulation timing

Key press time msec.

Key release time msec.

Function	Description
Key Press Time	Enter the length of time the key is held down. The minimum value is 0. The maximum is 640. The default is 20.
Key Release Time	Enter the time delay between keystrokes. The minimum value is 0. The maximum is 640. The default is 20.

Use this tab to configure the Software Developer’s Kit (SDK) functions, as well as enable and disable keystroking.



Software Developers Kit

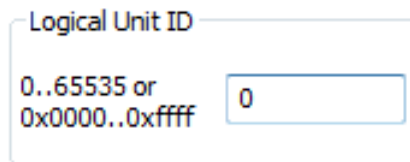
Function	Description
Disable Keystrokes for SDK (Halt Keyboard Send)	Check to disable keystroking. When keystroking or unsolicited serial out is disabled, all card data must be read via the SDK functions.

LED

Function	Description
Auto	Select this to make the device set the LED color.
Off	Select this to set the LED to off
Red	Select this to set the LED color to red .
Green	Select this to set the LED color to green .
Amber	Select this to set the color to amber .

Logical Unit ID

A user defined 16 bit Logic Unit ID to identify one device from another.

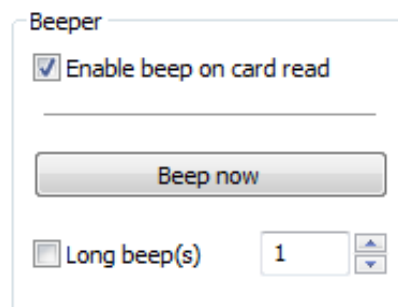


Logical Unit ID

0..65535 or
0x0000..0xffff

Beeper

Function	Description
Enable Beep on Card Read	Check this to set the device to beep when a card is read.
Beep Now	Press to listen to the beep the reader will provide when in use.
Long Beep(s)	Check the box to configure a long beep of 375 msec. By default the beep is set to a short beep of 125 msec



Beeper

Enable beep on card read

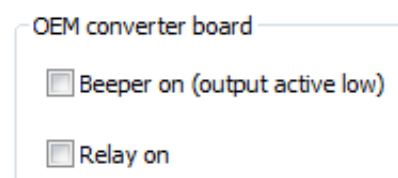
Beep now

Long beep(s)

The number value input area to the right of the Long Beep(s) box is designated for the number of beeps to produce when the device is in use.

OEM Converter Board

Function	Description
Beeper On (Output Active Low)	Check this to turn the device beeper on.
Relay On	Check this to activate the OEM board.



OEM converter board

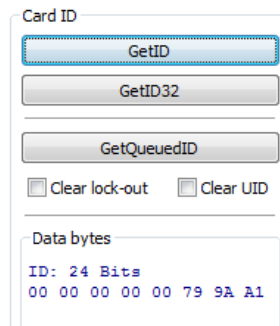
Beeper on (output active low)

Relay on

Card ID

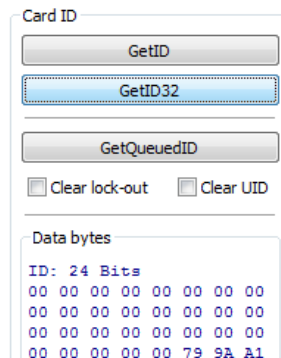
Function	Description
GETID	Click while scanning a card over the device. The ID displays under the button. This returns 64 bits maximum.
GETID (32)	Click while scanning a card over the device. The ID displays under the button. This returns 255 bits maximum.
GetQueuedID	Click to display the last card data read. This returns 255 bits maximum.
Clear Lockout	Check to clear the time remaining to allow the device to read the next card immediately.
Clear UID	If clearUID is set, the card and the over run counters will be cleared for the next read. If clearHold is set, the reader will be ready to read another card immediately.

GETID Data Display



The Most Significant Byte is first - E0.
The Least Significant Byte is last - 34.

GETID(32) Data Display



GetQueuedID Data Display

Card ID

GetID

GetID32

GetQueuedID

Clear lock-out Clear UID

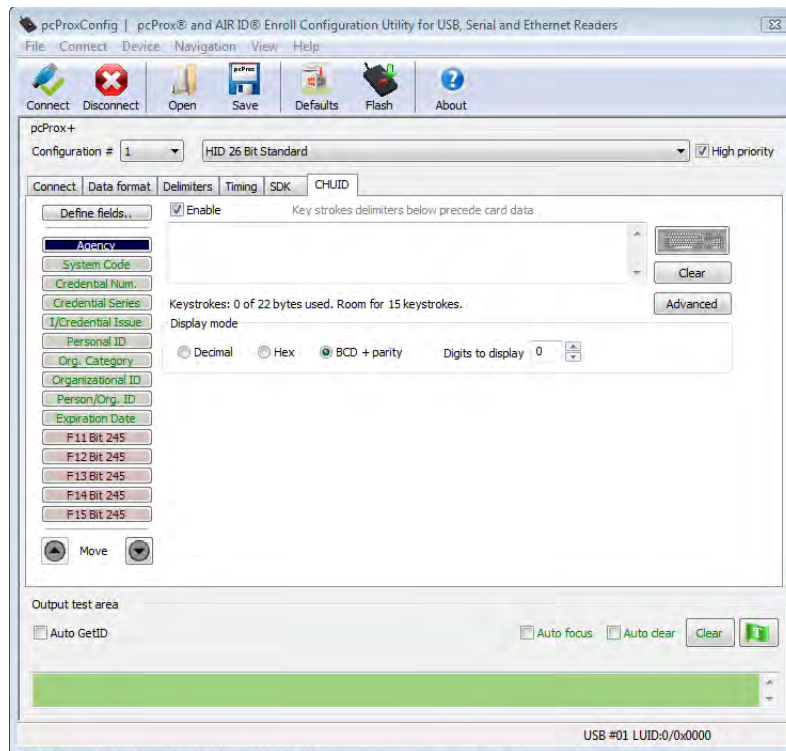
Data bytes

```
Queued ID: 64 Bits
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
E0 12 FF F8 00 44 B9 34
Age: 00:00:06 (127x48ms)
Over runs: 0
Lockout timer: 0
```

HH:MM:SS displays - 00:00:06

CHUID Tab

This tab allows manipulation of all fields on the Federal Information Processing Standard (FIPS) 201, or proximity cards. Use the red buttons to configure additional fields. The fields can be moved to change the order displayed in the binary bit pattern display.



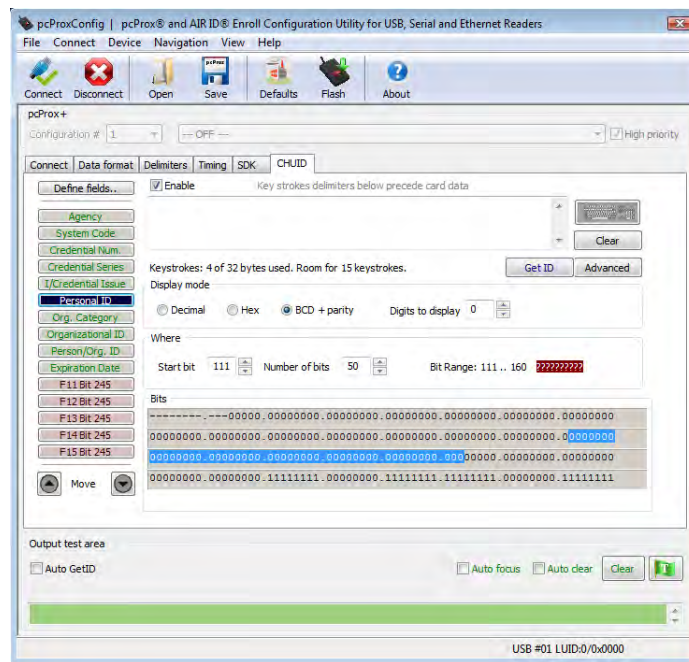
Function	Description
Define Fields	Click to select the number of source bits to define the fields. The correct type must be selected to allow for all card bits to be manipulated.
Enable	Check to enable the highlighted field. This allows the delimiters to be output and the corresponding card field to be processed and output. All green fields are enabled. All red fields are disabled.
Keyboard	Click to select key delimiters that are stored in the device's flash memory that precede card data output. Each field may have from 0 - 14 key strokes.
Clear	Click to clear keystrokes preceding the card data.
Decimal	Click to display the card field in decimal format.
Hex	Click to display the card field as a base 16 number in uppercase HEX 0 - 9 and A - F.
BCD w/ Parity	Displays the card data in binary coded decimal, where each 5 bits represent 1, 2, 4, 8, and parity. FASCN data is always odd parity.

Function	Description
Advanced	Click to display the binary bit pattern.
GetID	Click to display the binary bit pattern captured from the card.
Start Bit	Enter a number to define the left most significant starting bit for the field.
Bits	Enter the number of bits to add to the Start Bit to define the range of bits in the field.
Digits	This is the number of digits that will display in a selected field.
Up	Click to move the highlighted field up one position.
Down	Click to move the highlighted field down one position.

Advanced Button

This displays the bit ranges of the card.

Click each field button to display the location of the card binary data. In the example below, the Personal ID starts at bit 111, is 50 bits long, and is 10 digits. The Bit Range is 111 .. 160 and the card bit pattern is highlighted. This output format is displayed in binary coded decimal with parity (BCD with Parity). This is the 245 bit configuration. If any additional keystrokes were entered to precede the card data, click Clear to remove them.

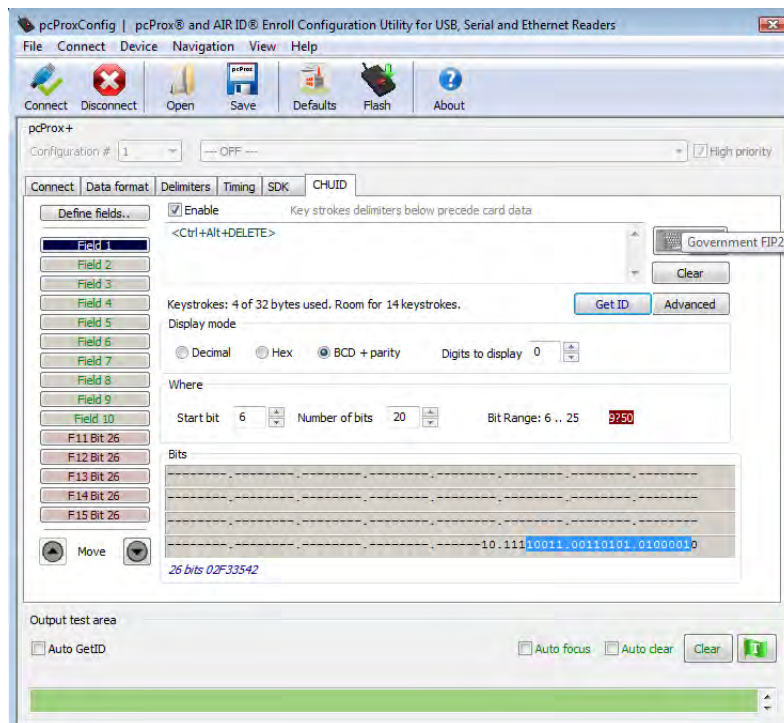


Note: The message that displays the number of bytes used and how much room for keystrokes above the **Advanced** button is determined by the device's flash memory. In this example the configuration is: "Keystrokes: 8 of 32 bytes used. Room for 14 keystrokes." Every field is 15 keystrokes maximum. All fields share 96 bytes.

The Bit Range that displays to the left of the binary bit pattern is the Start Bit field total + the Bits field total - 1.

Get ID

Click **GetID** and scan the card to display the output format of the FIPS 201 and proximity card and the interpretation display of the card data. Click **GetID** to define the fields to set up the device.



In this example, The Agency data starts at bit 11, is 16 bits long, and is 5 digits. The location of the agency data is highlighted in the binary bit pattern. The Bit Range is 6 .. 25.

The actual card data displays in blue below the binary bit pattern layout. The interpretation of the card data displays in red in the text field. The card data in blue will always be the same. The card data in red changes based on configuration settings flashed to the device.

Note: Click Clear to delete the red card data in the text field. A confirmation message will display.

The **Start Bit** changes the actual location of the selected field on the binary bit pattern.

Where

Start bit Number of bits Bit Range: 3 .. 22 ????

Bits

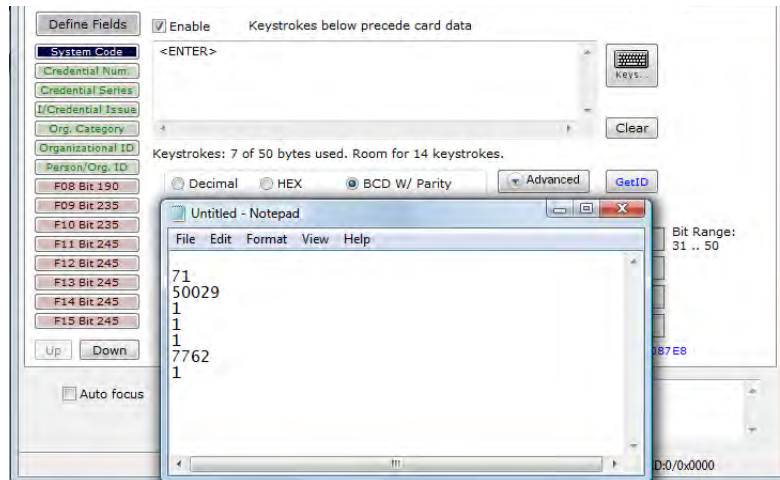
-----10.11110011.00110101.01000010

26 bits 02F33542

Note: The '????' that display to the right of the Digits field indicate the BCD parity is incorrect. Verify the correct field is selected.

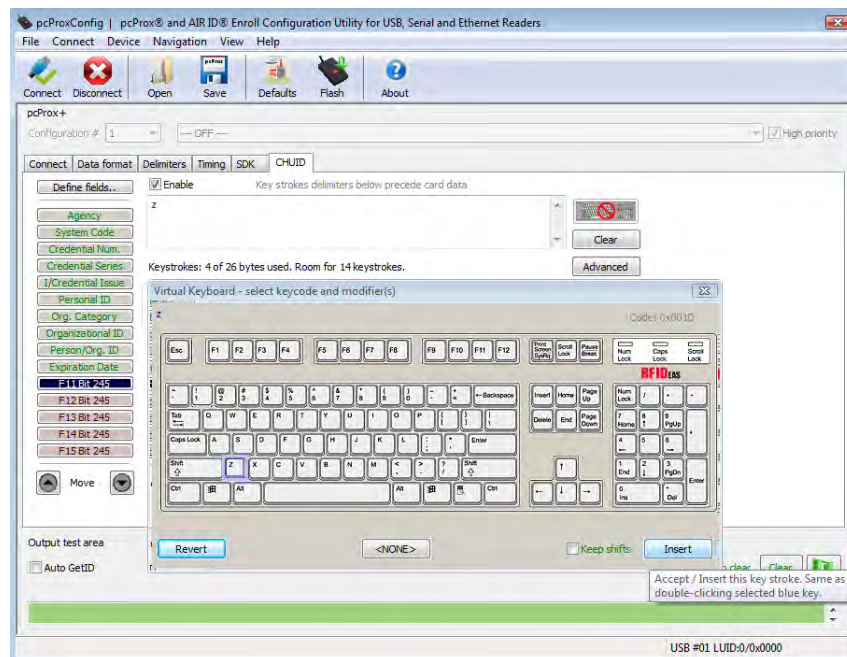
Change Fields Configuration

Click on the appropriate field button and uncheck **Enable** to remove field data from being displayed. In the example below, the Agency, Personal ID, and Expiration Date fields have been removed. Additional function keys display to configure more fields.



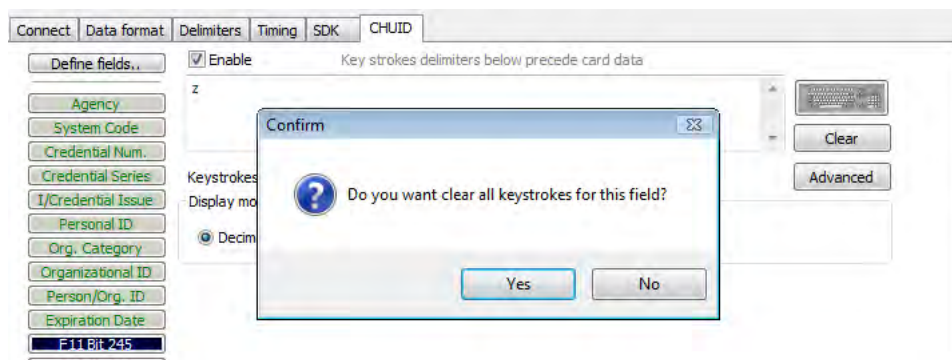
Assign Preceding Keystrokes

If **Enable** is checked for a field, specific keystrokes can be assigned to precede card data output.

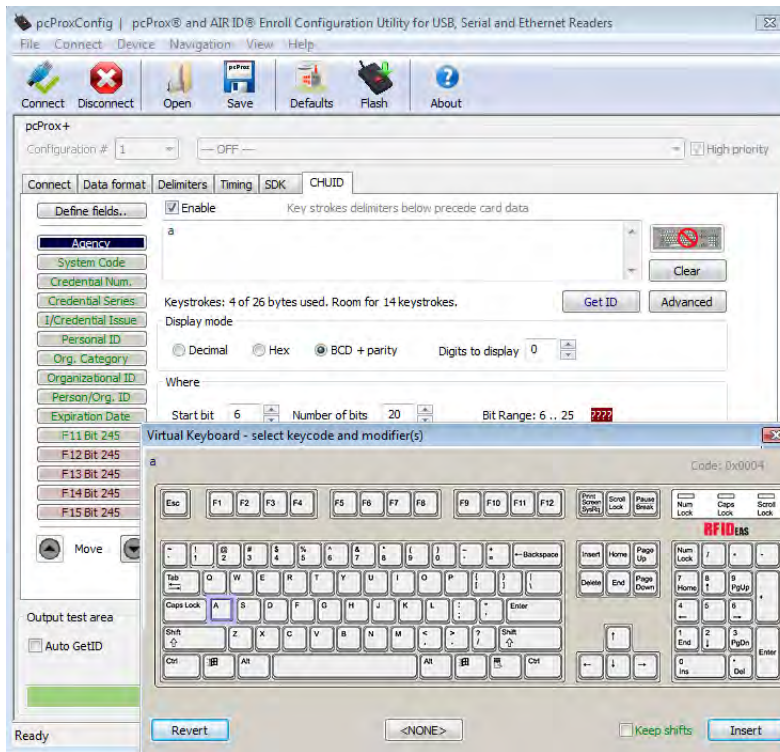


Note: The Scan Code output for the key selected displays above the list of keys.

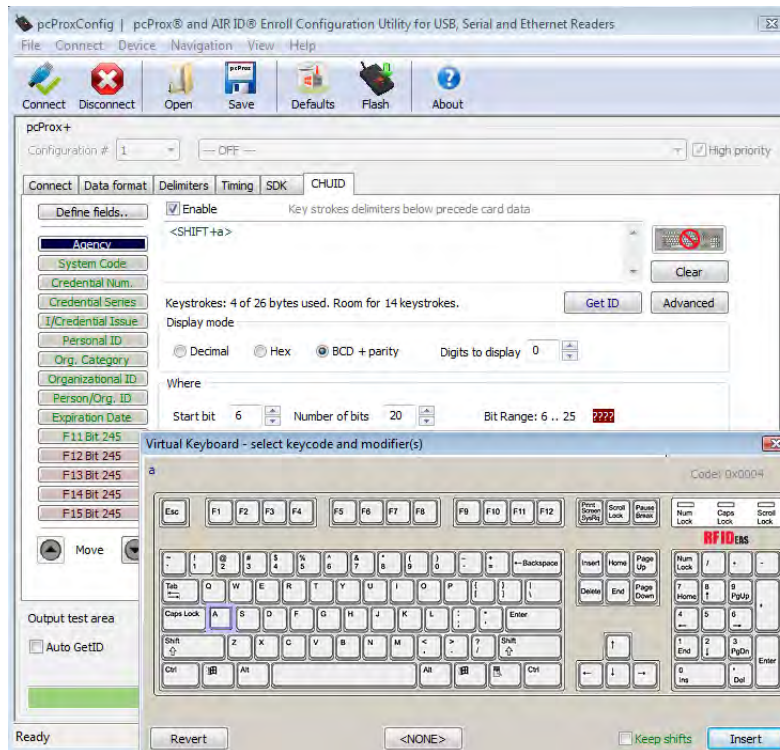
Click **Clear** to remove all preceding keystrokes as appropriate.



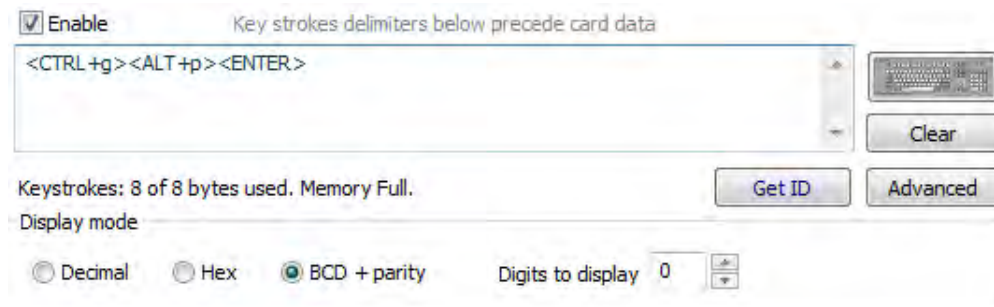
Each single keystroke entered to precede card data equals 1 byte of memory.



If any special character is selected with a keystroke, this equals 2 bytes of memory.

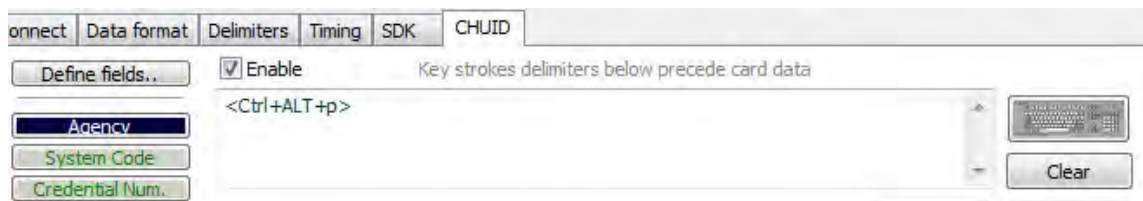


If all the keystrokes have been assigned to the fields, the following message displays:

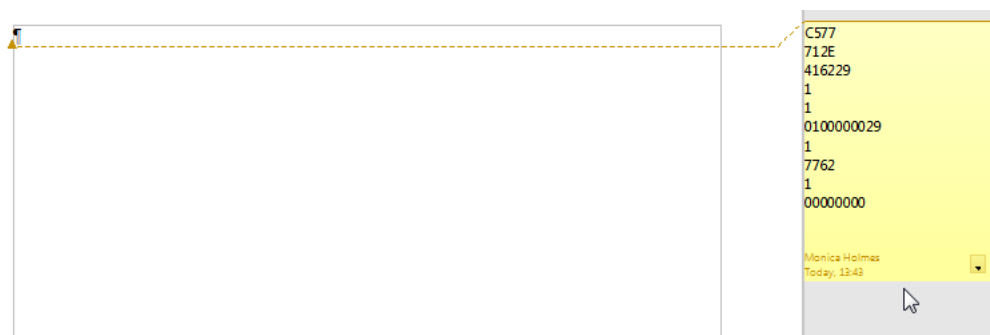


Depending on the active document/window, additional functionality can be assigned to a field. For example, if the card data is read in OpenOffice, the Note feature can also be assigned.

Select the appropriate field. Click the keyboard icon. Check **Left Control**. Check **Left Alt**. Click **n**. Click **Insert**. Click **Flash** to write this configuration to flash memory. Verify the active window is OpenOffice. Scan the card. The Notes function opens when the card is read.



The value assigned to the function key in the active document/window determines the output.



Note: This configuration utility creates a ComSpecPort.txt file and saves to the default directory. This file can be opened and deleted at will.

FIPS 201 Card Configuration

In order to configure a FIPS 201 card:

- Click **Advanced** to display the card data in the binary bit pattern to determine bit length and format
- Click **GetID** and present the card to the reader
- Define the fields to match the specific output
- Configure any additional fields as appropriate
- Flash the configuration to memory

The **Advanced** button displays:

- Start bit location
- Number of bits for a specific field
- Number of digits for the field
- Location of the field within the 245 bit range

ASCII Overview

ASCII Command Protocol (ACP) allows the user to talk directly to the device without a DLL or special application. The serial Prox communicates using ASCII commands. Printable ASCII commands at 9600 baud, no parity 1 stop bit, and no echo, can be sent to the device.

Note: USB devices that are virtual COM port do not need the baud rate set. The input is buffered by the device and executed when a carriage return (CR) or line feed (LF) is typed. The unit then parses the command and performs the operation, and displays the results or error code. “\r\nRF IDEas>” where \r represents a CR and \n represents a LF that displays on the command line.

All commands begin with the prefix rfid: and end with a Return key, CR or LF.

Determine the COM Port

Windows

Use device manager to display the COM ports. Open the serial COM port. If it is a CDC virtual port, open the newly installed device that was created.

Linux

Most Linux distributions include Minicom. Download putty (www.putty.org) to communicate with the serial device if Minicom is not available.

After the USB CDC device is enumerated on the Linux machine a device of either /dev/ttyACM0 or /dev/ttyACM1 is found in the /dev/directory. Minicom users may have to create a symbolic link from /dev/ttyACM0 to /dev/modem using the command `ln -s /dev/ttyACM0 /dev/modem` or `ln -s /dev/ttyACM1 /dev/modem`.

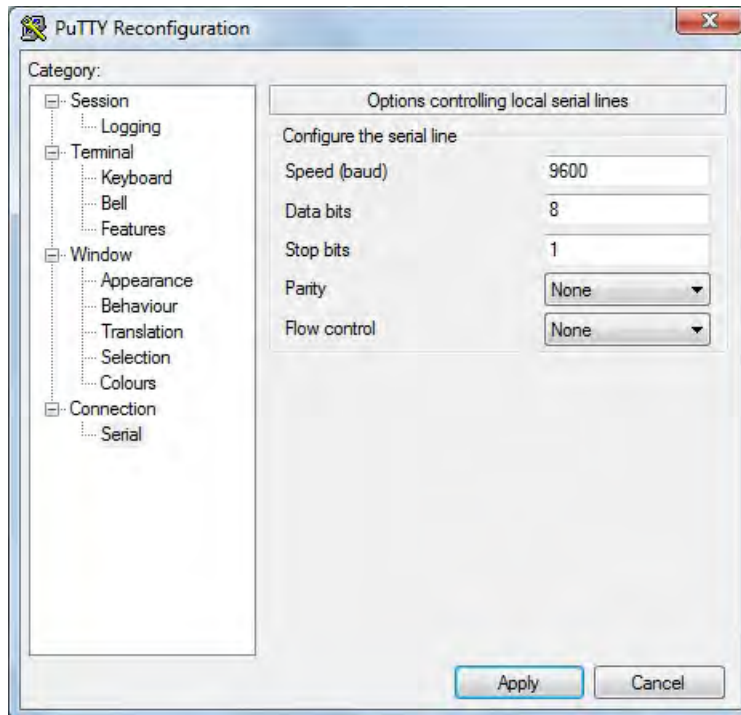
Mac OS X

The /dev/cu.usbmodemfa211 device is found on a Mac OS X . Use putty to communicate with this device.

Connect Serial Communications Program

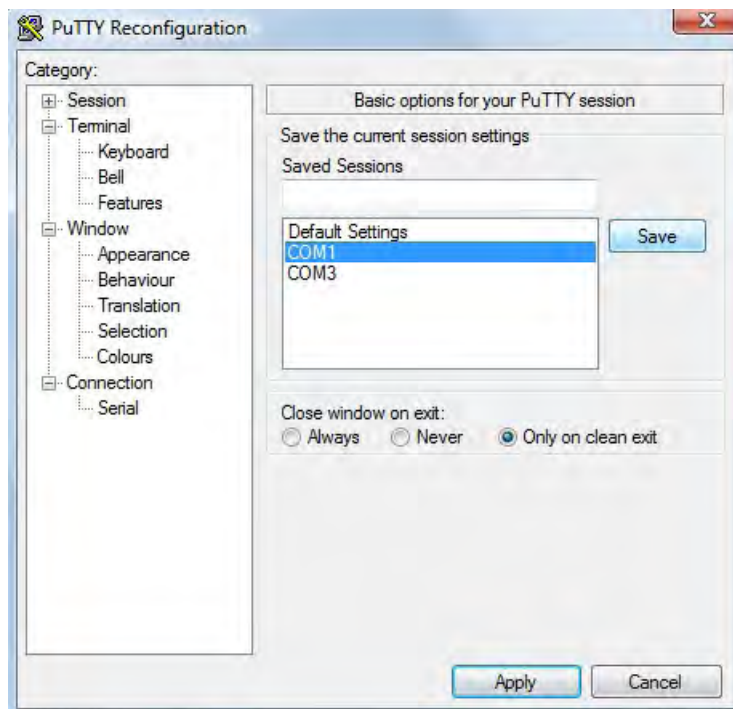
Open **putty.exe**. Click **Connection** ⇒ **Serial** and set the **Speed (baud)** to 9600, **Data bits** to 8, **Stop bits** to 1, and **Parity** to None. **Flow control** is not needed as there is no software or hardware handshaking.

Click **Session**.



Note: Use Hyper Terminal instead of putty with the XP operating system.

Highlight the appropriate session. Click **Save**. Click **Apply**.



PuTTY opens. Type **rfid:help** and press **Enter**. The Help command output displays. The complete list of Help command output is found in this section of the pcProx and AIR ID Enroll Configuration User Manual.

Command Structure

Commands are not case sensitive. Characters assigned to variables are case sensitive.

- All commands begin with a prefix string followed by one or more token strings with a period delimiter character between multiple tokens.
- Functions must end with a CR or LF.
- Variables can be assigned a value with an equal sign followed by the value or queried with a question mark.
- Any control characters other than CR, LF, and backspace terminate the command.
- The Escape key cancels a command.

The general syntax is:

```
PREFIX TOKEN { DELIMITER TOKEN } { { =Value } | { ? } }
```

The prefix string is rfid:

Command structure falls into one of three groups:

1. Perform a function.
2. Assign a variable.
3. Query a variable.

Perform a Function

A function performs an operation that may or may not display any results. A function may not be queried. An example of a function is to write the variable settings to flash memory using **rfid:cfg.write CR**.

Certain functions that display a value or series of values display the string between curly braces for easy parsing. For example, the **rfid:qid** function output displays:

```
{0x00BB,1,0x0000,80;0x000000801CD1931B2F14}
```

Assign a Variable

There are three types of variables:

1. Boolean
2. Integer
3. Character

Examples of Boolean Assignments

```
rfid:op.beep=0  
rfid:op.beep=true  
rfid:op.beep=False  
rfid:op.beep=F
```

Examples of Integer Assignment

```
rfid:out.led=0003  
rfid:out.led=3
```

Examples of Character Assignment

```
rfid:Delim.Chr.fac=':' CR  
rfid:Delim.Chr.fac='x3a' CR
```

Query a Variable

Query a single variable to display its current value.

- Booleans display as true or false.
- Integers display as 0..255 with leading zero suppression.
- Characters display as single quoted printable ASCII characters in the range 0x20..0x7E.
- Values from 0x00 .. 0x1F and 0x7F..0xFF will be with a leading backslash lowercase x and the two digit upper case hex number.
- The output of the variable displays between curly braces.

For example: **RF IDEas>rfid:out.led?**

```
{3}
```

Variables are set and stored in RAM and are lost when the utility is closed. Use the **cfg.write** function to write the RAM configuration to flash memory. Use the **cfg.read** function to read the flash memory.

Help Command

Help displays the commands followed by its data type and expected syntax. The table below displays the Help command output. The functions display in blue.

HELP COMMAND OUTPUT	
<code>rfid:cfg.read (Function)</code>	<code>rfid:disp.id.hex? =True False</code>
<code>rfid:cfg.reset(Function)</code>	<code>rfid:op.beep? =True False</code>
<code>rfid:cfg.write (Function)</code>	<code>rfid:help</code>
<code>rfid:chr.1? = 'A'..'z' '\x0D'</code>	<code>rfid:op.cont? =True False</code>
<code>rfid:chr.2? = 'A'..'z' '\x0D'</code>	<code>rfid:op.sdk? =True False</code>
<code>rfid:chr.3? = 'A'..'z' '\x0D'</code>	<code>rfid:out.beep? =True False</code>
<code>rfid:chr.count.lead? =0..15</code>	<code>rfid:out.led? =0..255</code>
<code>rfid:chr.count.trail? =0..15</code>	<code>rfid:out.relay? =True False</code>
<code>rfid:chr.eol? = 'A'..'z' '\x0D'</code>	<code>rfid:qid (Function)</code>
<code>rfid:chr.fac? = 'A'..'z' '\x0D'</code>	<code>rfid:qid.hold (Function)</code>
<code>rfid:chr.gone.1? = 'A'..'z' '\x0D'</code>	<code>rfid:qid.id (Function)</code>
<code>rfid:chr.gone.2? = 'A'..'z' '\x0D'</code>	<code>rfid:qid.id.hold (Function)</code>
<code>rfid:cmd.echo? =True False</code>	<code>rfid:time.hold? =0..255</code>
<code>rfid:cmd.prompt? =True False</code>	<code>rfid:time.lo? =0..255</code>
<code>rfid:dev.luid? =0x0000..0xFFFF</code>	<code>rfid:var (Function)</code>
<code>rfid:dev.part (Function)</code>	<code>rfid:wieg.id.bits? =0..255</code>
<code>rfid:dev.ver (Function)</code>	<code>rfid:wieg.inv.bits? =True False</code>
<code>rfid:disp.64bit? =True False</code>	<code>rfid:wieg.qual? =True False</code>
<code>rfid:disp.fac.digits? =0..255</code>	<code>rfid:wieg.qual.bits? =0..255</code>
<code>rfid:disp.fac.hex? =True False</code>	<code>rfid:wieg.rev.bits? =True False</code>
<code>rfid:disp.fac.send? =True False</code>	<code>rfid:wieg.rev.bytes? =True False</code>
<code>rfid:disp.fac.strip? =True False</code>	<code>rfid:wieg.strip.lead.bits? =0..15</code>
<code>rfid:disp.id.digits? =0..255</code>	<code>rfid:wieg.strip.trail.bits? =0..15</code>

Help Command Summary

rfid:cfg.read

This function tells the device to read the flash memory to RAM.

rfid:cfg.reset

This function resets the flash memory to the factory settings.

rfid:cfg.write

This function tells the device to write the RAM to flash memory.

rfid:dev.part

This function displays the part number of the device

rfid: var

This function tells the device to display the variable command output. This is similar to a .HWG file.

QID

The **rfid:qid** function exists in four forms:

- rfid:qid (Function)
- rfid:qid.hold (Function)
- rfid:qid.id (Function)
- rfid:qid.id.hold (Function)

Each **quid** function returns the same queued ID. The last 3 items control what is cleared after the function displays the output. The top line below is an example output string. The bottom line displays how this example is formatted.

EXAMPLE Output String: {0x1000,2,0x0000,80;0x000000801DD1910B2F04}
FORMAT of Output String: {AGE,OVERRUN,LOCKOUT TIME,BITCOUNT;ID}

AGE is the time in 48ms ticks that counts how long ago a card was scanned. This value count from 0 through 65535 displays in hex with "0x" hex notation. After 52.5 minutes the counter maxes out at 65535. The card data above shows this card was read 4,096 (0x1000 hex) x .048 = 196.608 seconds which equals 3 minutes and 16 seconds. The AGE counts until **65,535** (0xFFFF hex) and then maxes out. It will not roll over to zero. Use the qid.id function to clear the age counter.

OVERRUN is a counter from 0 through 255 displaying the number of cards scanned and over writes unread buffer contents. The device buffers one card. When a second card is read, the first card data is lost and the counter is set to one, meaning one card has overrun the buffer. The card data example above displays that 2 cards were read and the data from those cards was not transferred before reading this card.

LOCKOUT TIME is the number of 48ms ticks remaining until another card can be scanned. The card data above displays 10 times .048 which equals .48 seconds until the next card can be read.

BIT COUNT is the number of bits that follow 26 .. 255 and display as hex after the ';'. Notice the use of commas and semicolons. The card data example above shows that the ID contains **80** bits.

ID The card data above has 80 bits and is **0x000000801DD1910B2F04**.

QID.hold

This reads the card data as above and resets the hold lockout timer. Once the card data displays, a second card can be read immediately after without waiting for the lock out time period to expire.

QID.id

This reads the card data and also clears the age, overrun, and bit count after the values display.

QID.id.hold

This reads the card data and clears the ID variables and hold timer like both combined functions above.

SDK Command

Th `rfid:op.sdk=False` tells the device to display card data every time a card is scanned. If true, no card data displays. In the SDK mode, all keystroke or serial send data can be inhibited. The card data can be read using function `rfid:qid`.

Variable Command

The `var` command displays all variables. The command output can be captured and played back into the device. There must be a delay of several milliseconds after each character or the `pcProx` serial input buffer overflows.

VARIABLE COMMAND OUTPUT

```
rfid:chr.1='\x00'  
rfid:chr.2='\x00'  
rfid:chr.3='\x00'  
rfid:chr.count.lead=0  
rfid:chr.count.trail=0  
rfid:chr.eol='\x0D'  
rfid:chr.fac='\x00'  
rfid:chr.gone.1='\x00'  
rfid:chr.gone.2='\x00'  
rfid:cmd.echo=True  
rfid:cmd.prompt=True  
rfiddev.luid=0x1234  
rfid:disp.64bit=True  
rfid:disp.fac.digits=0  
rfid:disp.fac.hex=False  
rfid:disp.fac.send=False  
rfid:disp.fac.strip=False  
rfid:disp.id.digits=0  
rfid:disp.id.hex=False  
rfid:op.beep=True  
rfid:op.cont=False  
rfid:op.sdk=False  
rfid:out.beep=True|False  
rfid:out.led=255  
rfid:out.relay=True|False  
rfid:time.hold=20  
rfid:time.lo=24  
rfid:wieg.id.bits=80  
rfid:wieg.inv.bits=False  
rfid:wieg.qual=False  
rfid:wieg.qual.bits=80  
rfid:wieg.rev.bits=False  
rfid:wieg.rev.bytes=False  
rfid:wieg.strip.lead.bits=1  
rfid:wieg.strip.trail.bits=1
```

These five variables work together to display leading and trailing (pre and post) card data delimiters.

1. `rfid:chr.1='\x00'`
2. `rfid:chr.2='\x00'`
3. `rfid:chr.3='\x00'`
4. `rfid:chr.count.lead=0`
5. `rfid:chr.count.trail=0`

The first three commands identify the pre delimiter characters that can display. Three characters may be divided up as pre and/or post delimiters. Count.lead identifies how many of the three characters (chr.1 .. chr.3) display before the card data. For example, if count.lead is set to 1, only one character displays before the card data and chr.2 and chr.3 can be set as post delimiters. Then count.trail can have a value of 0, 1, or 2. If count.lead is 2, chr.1 and chr.2 are set as leading delimiters. Then only chr.3 can be set as a trailing delimiter. The same character can not be used for both a leading and trailing delimiter.

rfid:chr.eol='\x0D'

This command sends the End Of Line (EOL) character at the end of the card data. Typically a carriage return (CR) (0x0D) is used.

rfid:chr.fac=':'

This command sets a delimiter between the FAC and card data.

rfid:chr.gone.1='\x0A' and rfid:chr.gone.2='@'

These commands prompt the device to send the characters 'x0A' and '@' when the ID card is removed if they are not '00'.

rfid:cmd.echo=True

This command echoes user input when true and controls if backspace sends a space, backspace, space to erase the last character typed. If false, it is turned off for computer control. This value can be written to flash memory using `cfg.write`. It defaults to true on `cfg.reset`.

rfid:cmd.prompt=True

This command displays the prompt when true. If false, the prompt does not display. This value can be written to flash memory using `cfg.write`. It defaults to true on `cfg.reset`.

rfid:dev.luid=0x1234

This command sets the logical unit ID. A user-defined 2 byte value to identify this unit.

rfid:disp.64bit=False

This command uses 64 bit math to computer 64 bit decimal digits. This should always be kept on. If true, it uses 64 bit math.

rfid:disp.fac.digits=3

This command truncates or sets the FAC display leading zero.

rfid:disp.fac.hex=False

This command sends the FAC code in hex when true. If false, the FAC code is sent in decimal.

rfid:disp.fac.send=False

This command sends the FAC code if true. If false, the FAC code does not display.

rfid:disp.fac.strip=False

This command separates the FAC from the card data when true so it can be independently formatted for display. If false, the FAC code is not separated from the card data.

rfid:disp.id.digits=16

This command sets the digits so the left most significant digits will be truncated. For example, if the card data is 1234 and id.digits=3, then only 234 displays. If the card data = 8 formats the display width by truncating digits or adding leading zeros.

rfid:disp.id.hex=False

This command displays the card data as hexadecimal when true. If false, the card data displays as decimal.

rfid:op.beep=True

This command sets the device to beep on a successful card read when true. If false, the device will not beep even if the card is successfully read.

rfid:op.cont=False

This command sets the device to continuously read when true. This tells the device to read the same card data over and over while the card is on the device. If false, the device only sends the card data once.

rfid:op.sdk=False

This command stops the device from displaying the card data when true, so the qid or SDK API call must be used to get the card data. When true the device will send the data via keystrokes or serial depending on device type/model.

rfid:out.beep=False

This command makes the device beep when true. If false, the device will not beep. This is only available on OEM converter boards.

rfid:out.led=255

This command sets the variable and also sets the output LED color in RAM. Use cf.write to write this change to flash memory to persist across power cycles.

LED Value	Description
0	OFF
1	RED
2	GREEN
3	AMBER
4..254	Reserved
255	Controlled by the device

Rfid.out.relay= True

This command sets the output driver to ON (active low) when true. This is only available on OEM converter boards.

rfid:time.hold=20

This command sets how long in 48ms ticks the data is held for the active ID. This also controls how long the device keeps the LED green in 48 msec ticks. The default time is $20 * 0.048 = 0.960$ seconds.

Note: The **quid.hold** resets the internal timer this value initializes.

rfid:time.lo=24

This command sets how long in 48ms ticks the card device has to wait for no card in the RF field to begin accepting new card data. This prevents the same card data from being read over and over. If op.cont is true this value has no effect. The default time is $24 * 0.048 = 1.15$ seconds.

Note: The **quid.hold** resets the internal timer this value initializes, so that a new card can be read as soon as the data is transferred to the host computer.

rfid:wieg.id.bits=80

This command sets byte reversal and also defines the FAC bit size.

rfid:wieg.inv.bits=True

This command sets all ones to become zero in the Wiegand data. If false, all zeros are set to become a one.

rfid:wieg.qual=False

This command sets card reading filter to off. If true, card reading filter is on. This is related to the next command, .qual.bits=80.

rfid:wieg.qual.bits=80

This command sets the device to read only cards with this many bits. All other size cards are filtered out.

rfid:wieg.rev.bits=False

This command does not reverse all bits. If true, the least significant bits are swapped with the most significant bits.

rfid:wieg.rev.bytes=False

This command does not reverse all the bytes in the id.bits size field. If true, all bytes are reversed.

rfid:wieg.strip.lead.bits=1

This command strips 0 .. 15 bits from the most significant bits.

rfid:wieg.strip.trail.bits=1

This command strips 0 .. 15 bits from the least significant bits.

ACP Error Codes

Value	Display String	Description
1	{Error#1}	Illegal command. Wrong or Missing Prefix (rfid:).
2	{Error#2}	Input buffer exceeded. Too many character were typed without a CR or LF.
3	{Error#3}	Illegal operation, such as trying to query or assign a variable to a function or trying to use a variable as a function.
4	{Error#4}	Range Error. The value assigned to the variable does not make sense for its data type, such as try to assign 257 to a byte value.

Troubleshooting

If the device is not working or the following error message displays:



1. Check to be sure the device is plugged into the USB or RS-232 port. When the workstation is on and no card is being read, the LED is red. A valid proximity card causes the LED to turn green, provided the configuration is not set to only read certain bit lengths.
2. Only one COM port application can own the RS-232 port at a time. Make sure there is not another COM port application running. This prevents our software from seeing the device.
3. Verify the correct model and the software configuration screen agrees with the device attached.
4. Verify the port agrees with the workstation connector.
5. If the device still does not work, unplug it, remove 'General USB Device' using Windows **'Control Panel'** ⇒ **'Add/Remove'** Hardware. Then reboot the workstation. When the workstation boots up, re-attach the device USB and the OS should re-install the Windows driver automatically.

Change the release time to 1000 on the Timing tab for USB keystrokes to slow down the device. Open Notepad or Word and swipe a card to display the card data to see the actions of any non-printable symbols.

If the device does not read the card, contact the card manufacturer/vendor to verify that the card type is compatible with the device model.

Precautions

Do not mount the device directly on a metal surface. This could interfere with the RF signal and the operation of the device.

The device may not recognize valid cards in the presence of high RF fields. If current readings are erratic, take the following step:

- Move the equipment from any known transmitters nearby.

Contact Technical Support at 866.439.4884 for more information.

Standard 26 Bit Format Structure

There are several bits constructed together that comprise data sent from the proximity card to the device. There are numerous bit formats and lengths for proximity cards. The most popular is a 26 bit card format. The typical layout for this format is 24 bits of usable information as the first and last are parity bits to ensure data integrity.

The 26 bit format consists of 255 possible facility codes. Within each facility code there is a total of 65,535 unique card numbers.

The standard 26 bit Wiegand format is H10301. It is binary encoded data. The format consists of 2 parity bits, 8 bit facility code (F) and 16 bit card number fields (B). This format displays below.

```
PFFFFFFFFBBBBBBBBBBBBBBBBBP  
XXXXXXXXXXXXX.....  
.....XXXXXXXXXXXXXO
```

- Bit Coding
- P = Parity
 - O = Odd Parity
 - E = Even Parity
 - X = Parity mask
 - F = Facility code, range = 0 to 255
 - B = Card Number, range = 0 to 65,535

In general, the 26 bit format is the industry standard format. Primary benefits of this include:

- Open format
- Convenient to order
- Universal access control panel acceptance

The sale of this format is not limited to any one company yet the range of card numbers available in this format is limited. There is a potential for card numbers to be duplicated.

Please go to www.RFIDeas.com and follow the **Support** ⇒ **Learning Center** ⇒ **Proximity Card Formats** link for more details. The card manufacturer may also have additional details about the card format.

Complex Passwords

It is possible with certain limitations, to use the proximity token as a password for an application or operating system log on. The unique card bit-stream converted to either decimal or hexadecimal becomes the entire or a portion of the password. Enroll this card data to the password of the operating system application for the user.

Since the proximity token has no read/write memory there is no way to change this or write alphanumeric characters such as a user name to the proximity token. Some examples are shown below. Please see RF IDEas AIR ID Playback Starter Kit or call the Sales Department if this capability is needed.

Several companies have adopted a policy that requires users to change their password every xx number of days to increase security. The PIN is the portion of the password the user changes every xx number of days. Since the card data is completely numeric, any alpha and upper/lower case letter constraints are handled in the user supplied PIN.

A two-factor authentication system is made up of:

1. Card ID data
2. Personal Identification Number (PIN)

The device may be configured to allow operation under either a one or two-factor authentication system.

One-Factor

In a one-factor system, the user simply scans the ID card. The device may be configured to add TAB keystrokes ahead of the data as well as a TAB or ENTER keystroke after the card data.

Two-Factor

The two-factor approach is especially useful when insisting on password construction rules or periodic changing of passwords.

In a two-factor system, the user may enter the PIN either before or after the card data. If the user adds the PIN before the card data, the device may be configured to append the ENTER keystroke.

Pre and Post Characters

There are some additional measures that can be taken to make it more difficult for unauthorized users to reproduce the password.

Add additional keystroke characters to the card information that are difficult to re-produce while configuring the data. These additional characters are labeled special1, special2, and special3 on the Delimiters character menu selections.

A	AIR ID Playback 62, 64	
	ACP Error Codes 58	
	ASCII 6, 7, 10, 17, 18, 47	
	Assign Preceding Keystrokes 42	
	Auto Clear 25	
	Auto Correct 20	
	Auto Focus 25	
	Auto GetID 24	
B	Beep 21	
	Beeper 34	
	Button Bar 15	
	About 15, 16	
	Connect 15, 16	
	Defaults 15, 18	
	Disconnect 15, 17	
	Flash 15, 19	
	Open 15, 17	
	Save 15, 18	
C	Card Compatibility 10	
	Card Data Hold 31	
	Change Fields Configuration 41	
	Clear Button 25	
	Clone Configuration 20	
	COM 11, 22-23, 59	
	Complex Passwords 62	
	Connectors 9	
	Continuous Read 31	
D	Data Delimiters 29-30, 37	
	Device List 22	
E	Emulate Prox Pro 28	
	Ethernet 20, 22, 23	
F	FAC 27-29	
	FAC Digits 27	
	Factory Defaults 20	
	FIPS 201 37, 39, 46	
G	GETID 35, 38-39	
	GetQueuedID 35	
H	.HWG 15, 17-18, 20	
I	ID Digits 27	
K	Key Press Time 32	
	Key Release Time 32	
L	LED 33, 57, 59	
	Legacy Settings 28	
	Lock-Out Time 31	
	Logical Unit ID 34	
M	Manufacturer Card Compatibility (See Card Compatibility)	
	MIFARE 28	
	Minimum System Requirements 32	
O	OEM 34	
	One-Factor Security 62	
	Output Test Area 24	
P	pcProxConfig Manual 21	
R	RS-232 8-11, 22-23, 59	
S	SDK 6, 7, 10, 33	
	Serial 20	
	Shipping Configuration 20	
	Start Bit 28, 40	
T	Termination Keystroke 30	
	Test Button 25	
	Tool Tip Balloon 21	
	Two-Factor Security 62	
U	USB 8-11, 20, 22-23, 31-32, 47	
W	Wiegand 10, 26, 28	
	Wiegand Bits 28	
	Wiegand Bytes 28	
	Wiegand to Keystroke Data 26	

Other Products & Accessories



Software Developer's Kit

Allows independent developer's to use their application to read proximity access badge Read ID data of more than 1 billion cards in the field



PVC Label Proximity Card

Credit card size with paper release liner, 500 cards per box



Complete selection of various manufacturers proximity cards, labels and key fobs. Marked with data code and ID number, available in several Wiegand formats



AIR ID Read/Write Contactless SDK

Reads and writes directly to the smart cards



AIR ID Writer and Playback

Desktop read-only for iCLASS and NXP and smart cards



AIR ID Playback Starter Kit

Plays back card sector data in ASCII or keystrokes



pcProx Sonar

Presence detector configured as a keyboard



PS/2 to USB Power Tap

Powers a USB RF IDEas device from a PS/2 port



Mounting Brackets

Further adjust the standard mounting of the device angle

RF IDEas Inc.
© 2011 RF IDEas. All rights reserved.

Specifications subject to change without notice.

Windows, Macintosh, Solaris, Sun Ray and Linux are trademarks of their respective companies.
All other trademarks, service marks and product or service names are property of their respective owners.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. RF IDEas assumes no responsibility with regard to the performance or use of these products.
All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.

Please feel free to call, e-mail or visit our web site for a full list of applications, products, configuration options, supported cards and form factor specifications. Our web site includes application videos, support materials, case studies and detailed information about our product line.

Every effort has been made to ensure that the information in this manual is accurate. RF IDEas is not responsible for printing or clerical errors.