

IDWEDGEKB™ USERS GUIDE

USB KEYBOARD – USB VIRTUAL COM PORT
ID SCANNER

VERSION 3.5 SOFTWARE



COPYRIGHT

Copyright 2002 - 2022 TokenWorks, Inc. Printed in the USA

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TokenWorks, Inc.

www.TokenWorks.com (Company Web Site),

www.IDScanner.com (Product Web site)

www.IDScanner.us (Support Site)

TokenWorks®, CardVisor®, CardTool®, IDVisor®, AgeVisor® and IDWedge® are registered trademarks of TokenWorks, Inc. IDWedgeKB and IDWedgeBT is a TokenWorks Inc trademark.

Copyright	2
Introduction.....	6
Document overview	6
Support Site – IDScanner.us	6
HARDWARE OVERVIEW	7
<i>Power</i>	<i>7</i>
<i>Function Button/indicator</i>	<i>8</i>
<i>LED Status</i>	<i>8</i>
How to Scan or Swipe License	9
Software overview - Formula and Configuration Files.....	10
<i>Reading/Writing Formula and Configuration files.....</i>	<i>10</i>
<i>Configuration file – Config.txt.....</i>	<i>12</i>
Version	12
CreditCard	12
1DBarcode.....	12
MagTrackCard	13
MidInitial.....	13
FirstInitial	13
5DigitZip	13
DobYYOnly	13
ExpYYOnly	13
ScanMsg	13
DisableMagDL_CC	13
USBKeyboard	14
USBSerialPort	14
CCRaw	14

CharDelay	14
<i>Formula Files – Driverslicense, creditcard, oneDbar, & Magnetic</i>	15
Driverslicense.txt	16
CreditCard.txt.....	17
OneDbar.txt	17
Magnetic.txt.....	18
<i>Control keys</i>	21
<i>Control COMBINATION keys</i>	22
<i>Inserting Static text</i>	23
Formula and Config File Testing and Problem.txt.....	23
<i>Type of tests and example of diagnostic messages</i>	26
Config.txt test and message.....	26
Card Formula files test and message	27
Magnetic.txt card formula test and message	27
<i>Accessing the file system for editing</i>	28
<i>Editing and clearing Problem.txt</i>	29
Additional Information and Examples.....	30
Serial Port Mode	31
<i>Serial Port Mode DL/ID Card CSV string</i>	31
<i>Serial Port Mode Credit Card CSV string</i>	32
<i>Serial Port Mode Credit Card CCRaw=True</i>	34
<i>Serial Port Mode Magnetic cards CSV string</i>	35
<i>Examples of Serial Port Mode Output 1D BarCode messages</i>	37
Document Change Log	38
Appendix A Drivers License Data Stripe Technology	39
Appendix B IDWedgeKB Dimensions and Specs	41
Appendix C – Optional KIOSK Cable	43

Appendix D – Hexadecimal Key code chart	44
--	-----------

INTRODUCTION

The IDWedgeKB is a barcode/magnetic card scanner that supports a USB connection to a host computer and functions as a USB Keyboard device or Virtual Serial Port ID ID/Card Scanner (depending on configuration). Cards are scanned and parsed into fields, and these fields are sent as keyboard strokes to the host via the USB keyboard driver or as a fixed order, comma delimited data string via the USB Virtual Com Port. The configuration of the device, keyboard formulas are defined by files stored in the IDWedgeKB Flash memory.

DOCUMENT OVERVIEW

This document will describe the basic functionality of the IDWedgeKB and how to configure the application parameters and formulas used to control the output of data from the scanner for the HID-Keyboard mode and Serial Port mode.

SUPPORT SITE – IDSCANNER.US

Please visit IDScanner.us for latest application notes, software, troubleshooting help for IDWedgeKB. We use this site to support customers so please review this material.

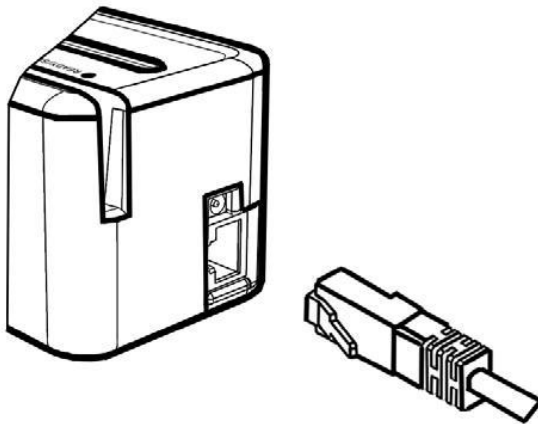
HARDWARE OVERVIEW

The picture below shows the side view of the IDWedgeKB scanner and the location of the function button/LED indicator and RJ50 USB connector.



POWER

The USB cable supplies power to the Scanner. The Scanner should be connected to a standard USB 2.0 or higher port (i.e. supply 500Ma of power). Note: Some older mobile laptops/tablets do not have standard USB ports which can drive 500 mA. There is no on/off button, the USB cable connected to the PC is the equivalent to the powered on state, and removing the cable powers off the Scanner.



The supplied USB – RJ50 cable provides communication and power to the IDWedgeKB scanner.

FUNCTION BUTTON/INDICATOR

The Function button/indicator is used to force the scanner into USB flash drive mode and allow access the configuration memory of the Scanner. If you press and hold this button during power up, you will put the device into USB Flash drive mode (i.e. USB thumb drive). The configuration and formulas files can be edited via the File System as well as new firmware copied to the drive memory of the device.

LED STATUS

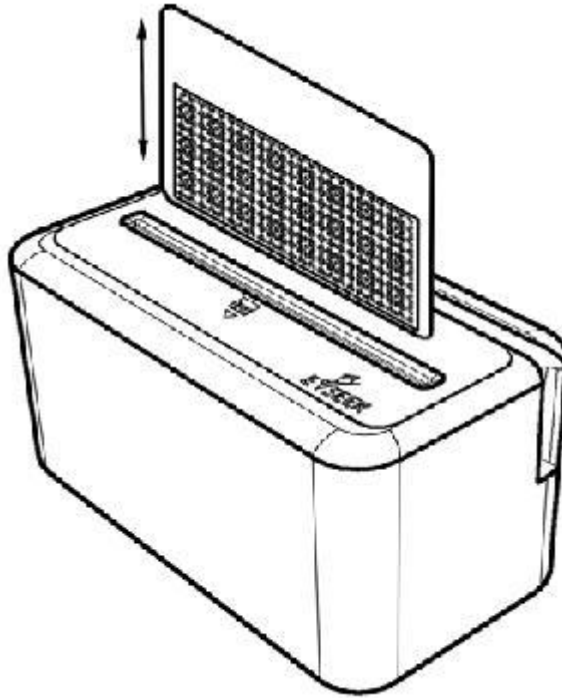
This LED inside the translucent Function Button will flash when the IDWedgeKB is powered on. The colors and rate of flash determines which mode it is in:

LED	Mode	Note
Green Flashing – 3 flash/sec	Boot loader Drive	Special Mode for firmware recovery
Green Flashing – 10 flash/sec	Flash Thumb Drive	Edit configuration and formula files
Blue Flashing – 3 flash/sec	USB Keyboard	Normal Operating Mode
White Flashing 3 flash/sec	USB Serial Com port	Normal Operating Mode

HOW TO SCAN OR SWIPE LICENSE

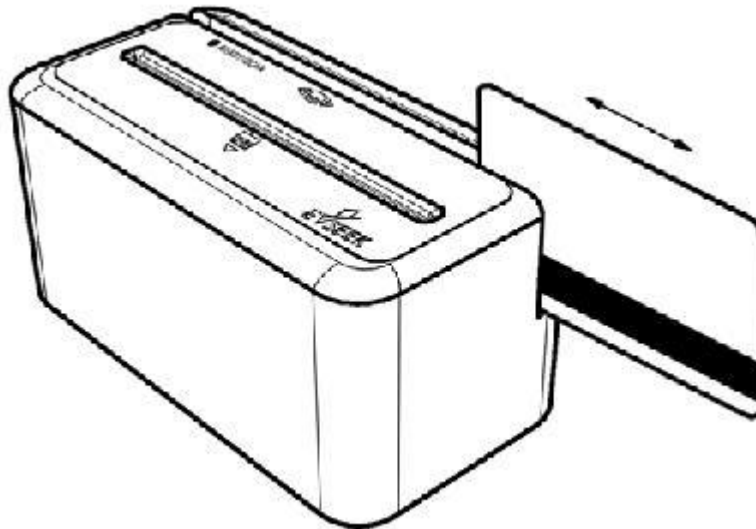
2D Barcode Scanning

For states like NY with 2D bar code and Military IDs, insert bar code with bar code facing slot as shown.



Magnetic Stripe Swiping

For states like California with a Magnetic Stripe, swipe license with magnetic stripe facing slot as shown.



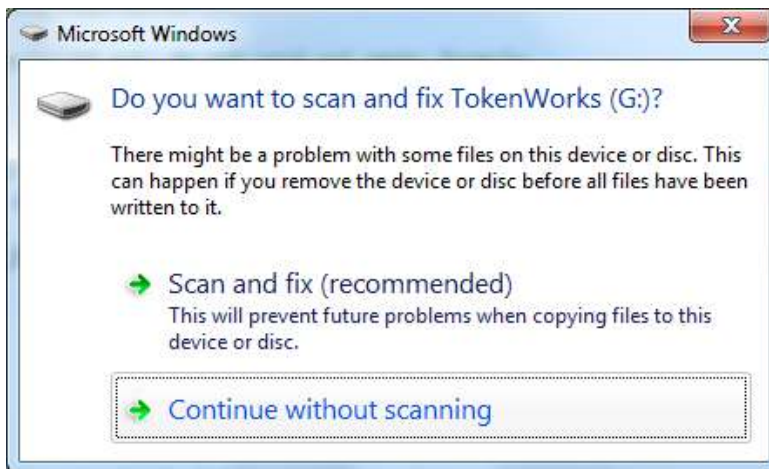
SOFTWARE OVERVIEW - FORMULA AND CONFIGURATION FILES

The formula and the default configuration files are created by the IDWedgeKB. Users then can modify these files to customize the behavior of IDWedgeKB. If a file is erased, the default version of the file will be re-created during next power cycle. Additionally, there are automatic test performed on these files during power up, to ensure the files adhere to some basic formatting rules. If an error is detected, a diagnostic file called: Problem.txt will be created and contain the error condition to be fixed, and the name of the file that contains the error. Once the file's error(s) have been corrected, the Problem.txt file will be automatically deleted during the next power cycle. This diagnostic file will remain, until all errors have been cleared.

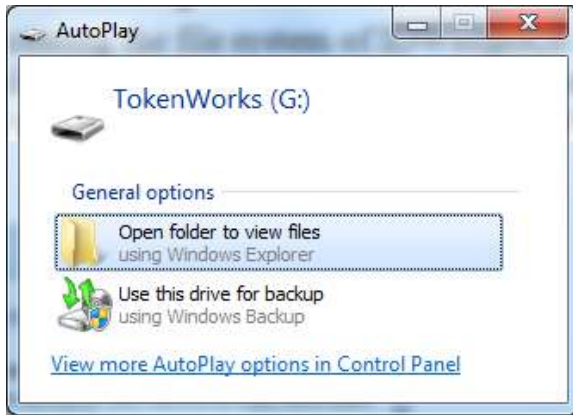
READING/WRTING FORMULA AND CONFIGURATION FILES

To Access the Formula and Configuration files, you must hold the function button during power up. Once you have connected via USB cable and can explore the contents of the scanner memory, you will be able to read and write the Formula and *Config.txt* file. You can use any SIMPLE text editor like note.exe to change items in the file and save them to the memory. DO NOT USE MS WORD AS IT WILL ADD FORMATING INFORMATION WHICH WILL CORRUPT THE FILE FORMAT.

Press and hold the function button after you connect the USB cable to a PC and you will see the new drive dialog box below.

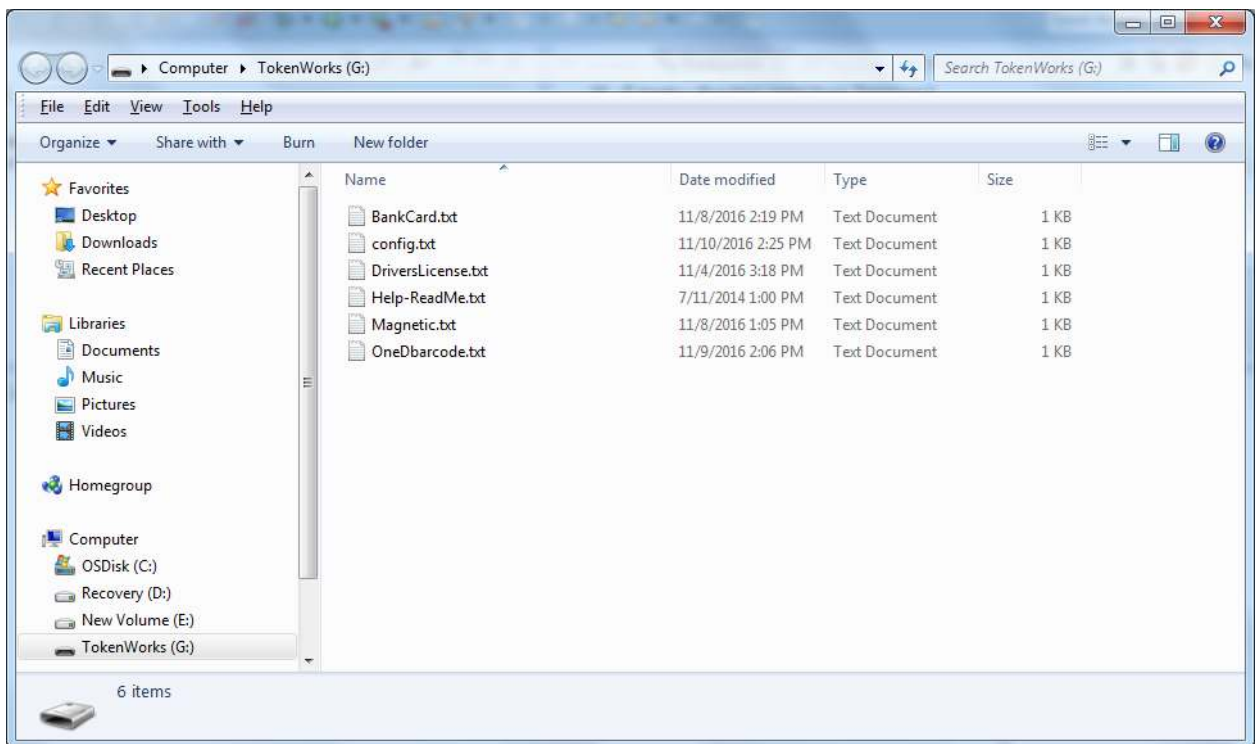


Select Continue without scanning, and you will see:



Open with Windows Explorer.

The file system normally contains 6 files as seen in this screen shot:



The above files control the behavior of IDWedgeKB.

CONFIGURATION FILE – CONFIG.TXT

The IDWedgeKB uses a configuration file called *Config.txt* to control some basic hardware parameters. If the file gets erased, it will be re-created automatically using factory default settings during next power on. If the file contains editing errors or is corrupted and cannot be opened, then a file called *Problem.txt* will be written to the SD card. Once the errors have been corrected and the file can be read, the *Problem.txt* file will be automatically erased. Tip – If unable to fix config.txt, just erase config.txt to return to factory default state.

Below is an example of the contents of the default *Config.txt* file followed by an explanation of what each field does.

```
<Version=3.5>
<CreditCard=True>
<1Dbarcode=True>
<MagTrackCard=True>
<MidInitial=False>
<FirstInitial=False>
<5DigitZip=False>
<DobYYOnly=False>
<ExpYYOnly=False>
<ScanMsg=True>
<DisableMagDL_CC=False>
<USBKeyboard=True>
<USBSerialPort=False>
<CCRaw=False>
<CharDelay=0>
```

VERSION

This is the firmware version, e.g. 3.5 and reflects the latest version of code.

CREDITCARD

CreditCard Controls the output of Bank Card scans, if enabled, then Magstripe cards that are identified as Bank Cards will be parsed and the fields will be available for use in CreditCard.txt formulas. If CreditCard is disabled, then CreditCard.txt formulas will be disabled.

1DBARCODE

1DBarcode controls the output of Code 39 and Code128 barcodes, if enabled then OneDbar.txt formula will be active. If 1DBarcode is disabled, then there will be no output when a Code 39 or Code128 barcode is scanned.

MAGTRACKCARD

MagTrackCard controls the output of Magstripe cards like Student Ids, AAA cards, etc. If enabled, the Magstripe.txt formulas will be active if a magnetic card is identified. If disabled, then Magstripe.txt formulas will be disabled.

MIDINITIAL

The MidInitial controls the output of the Middle name field, if set to true then only the middle initial will be output, if set to false then the whole middle name (if available) will be output.

FIRSTINITIAL

The FirstInitial controls the output of the first name field, if set to true then only the first initial will be output, if set to false then the whole first name will be output.

5DIGITZIP

The 5DigitZip controls the output of the Zip Code field, if set to true only the first 5 digits of the zip code will be output. If set to false then the entire zip code field will be output, this can range from 5 to 10 digits.

DOBYYONLY

The DobYYOnly variable only sends out the last two digits of the YYYY DOB field. If true then DOB_yy outputs YY, if false DOB_yy outputs YYYY

EXPYYONLY

The ExpYYOnly variable only sends out the last two digits of the YYYY EXP field. If true then EXP_yy outputs YY, if false EXP_yy outputs YYYY

SCANMSG

The ScanMsg controls the output of card error messages, when set to true, you will see messages like “scan again” or “decode error”. These messages can be helpful when developing formulas, but will cause extra operator effort when attempting to fill out forms.

DISABLEMAGDL_CC

The DisableMagDL_CC controls the behavior of the Magstripe parsing. When set to true, all Magstripe cards will be subject to the magnetic parsing formulas (See 3.4 Magnetic.txt). When set to false cards will be identified as either Drivers Licenses or Credit cards, and subject to the appropriate formulas, if enabled. For normal operation of Driverslicense.txt formulas, and CreditCard.txt formulas, this variable must be set to false.

USBKEYBOARD

The USBKeyboard, when set to true, puts the device into Keyboard mode, and uses the formula files to control the output of keyboard characters to a form. This variable is mutually exclusive to USBSerialPort, which means that USBKeyboard and USBSerialPort cannot both be true or false simultaneously.

USBSerialPort

The USBSerialPort, when set to true, puts the device into Serial Port Profile, and uses a fixed Comma Separated Value (CSV) string containing all the fields, as the formula. If the USBKeyboard and USBSerialPort are both True or both False, then the USB flash drive will be activated when the IDWedgeKB is plugged into a PC so the user can fix this error in the config.txt file. Note: Please visit Serial port section of manual for instructions on installing virtual USB com port drivers.

CCRAW

The CCRaw, when set to true, is only active when USBSerialPort is also set to True, and USBKeyboard is set to False, and Credit Card is to True and DisableMagDL_CC is False. When configured in this manner, and a credit card is scanned, the output will be the raw data of Track1 and Track2, plus the normal sentinels associated with Serial Port communications. Note: Please visit Serial port section of manual for details.

CHARDELAY

The CharDelay controls the amount of time to wait between the transmissions of characters, the range is 1-9 and this number is multiplied by 10Ms for a range of 10 to 90 Ms. A setting of 0 defaults to 1Ms

Note how there is no space between the equal sign and the value:

<CharDelay=0>

If a space is entered after the equals sign, this will cause the Problem.txt file to be generated with the message: *Reading CharDelay value of 0-9 fails in Config.txt*, and will have to be corrected.

FORMULA FILES – DRIVERSLICENSE, CREDITCARD, ONEDBAR, & MAGNETIC

The Formula used to send fields to the host is stored in a file on the scanner memory and are only used for USBKeyboard mode (except the magnetic formula which is also used in serial mode).

To Access the Config.txt file, you must boot the IDWedgeKB while holding the button as described in section 1.2.2. Each of the 4 card type supported has its own formula stored in its own formula file. The 4 formula files are: *Driverslicenses.txt* for ID/DL cards, *CreditCard.txt* for Credit cards and *OneDbar.txt* for 1D Barcode cards and *Magnetic.txt* for Magnetic stripe cards.

The concept behind the formula is that each parsed field on a card is represented by a single character. Control characters are enclosed with curly braces and appear in between each field in the formula. Below is a simple formula for filling out an address from a Drivers license:

F{TAB}L{TAB}A{TAB}C{TAB}S{TAB}Z{ENTER}

This formula would fill out a form that had text boxes for each item. The **F** represents the first name; the **L** represents the last name, followed by **A** for address, **C** for city, **S** for state and **Z** for zip code. The {TAB} in between each field moves the cursor to the next text box on a form, and the {ENTER} would act as if the user hit the Enter key to submit the data.

NOTE: There must be at least 1 valid field in each formula to create an output message and a field must be the first entry. The formula must occupy a single line of text, unless it is the Magnetic.txt formula.

The formula file has a maximum length of 500 characters. If your formula is greater than 500 characters it may cause problems. For reference, the formula above contains 38 characters.

DRIVERSLICENSE.TXT

The Drivers license formula uses capital letter to represent the fields on a drivers license, ether barcode or magstripe cards;

F – First name
M – Middle name
L – Last name
T – Title *

A - Address
C - City
S - State
Z – Zip code
B – DOB (MMDDYYYY)
E – Expiration (YYYYMMDD)
D – License number
H – Height
I – Eye color
O – Class *

P – Hair
U – Weight
V - Endorsements *

X – Sex or Gender
N – Scanner S/N
? - Restrictions
& - Document Issue Date **
@ - Country (USA or CAN) **
- 2nd Address field **
G – Real ID **

Q - DOB_mm
W - DOB_dd
Y - DOB_yy (yyyy format via config.txt variable)
J - EXP_mm
K - EXP_dd
% - EXP_yy (yyyy format via config.txt variable)

Example:

```
F{TAB}L{ENTER}C[ ]S[, ]Z{ENTER}
```

* Military IDs (CaC) returns Rank in Title field, Pay Grade in Class, and ID# (base10 format) in Endorsement

** (Available on PDF417 2D Barcodes only)

NOTE: Users can use the Q, W, Y field to create any format for the DOB field, and J,K, % for the EXP field.

CREDITCARD.TXT

The Credit Card formula uses lower case letters to represent the fields on a credit card.

a = Track1 Raw data
b = Track2 Raw data
f = first name,
l = last name
p = PAN or credit card number
y = ExpYY,
m = ExpMM
n = Scanner S/N

Example:

`f { TAB } l { TAB } p { ENTER }`

NOTE: *Raw data means all the information on a track including the start and end sentinels in ASCII format .*

ONEDBAR.TXT

The OneDbar formula uses lower case letters to represent the fields on a 1D barcode card.

w = 1D Barcode
n = Scanner S/N

Example:

`w { ENTER }`

MAGNETIC.TXT

The Magnetic Formula used to send fields to the host and is stored in a file on the SD card. To Access the Config.txt file, you must boot the IDWedgeKB while holding the battery button.

The magnetic parsing formula will allow for two tracks to be parsed and two fields of specific lengths and specific offsets to be parsed per track for a total of 4 fields. Each field is represented by a track number, and offset starting from zero and a length. There are three tracks and each track has 2 fields associated with it, for a total of 6 field designators

The total number of possible field is 6 and each field is mapped to a lower case letter (field designator):

- q** – Track1 Field1
- r** – Track1 Field2
- s** – Track2 Field1
- t** – Track2 Field2
- u** – Track3 Field1
- v** – Track3 Field2
- n** – Scanner S/N

The *Magnetic.txt* must have two lines of data separated by a carriage return; the first line defines the track number, offset, and length associated with each field. These values are stored as a comma separated string of 10 integer values. If you do not need all of the 4 possible fields, then populate the unused track, field and offset with zeros.

Example of first line of Magnetic.txt: 1,2,4,6,9,2,1,5,6,5

(Track 1, offset1 = 2, length1 = 4, offset2 = 6, length9 = 5, track 2, offset1 = 1, length1 = 5, offset2 = 6, length2 = 5).

The second line of the Magnetic.txt uses the lowercase field designators, along with the control words, to place the card fields in the form at the desired location.

Example of second line of Magnetic.txt:

q{TAB}r{TAB}s{TAB}t{ENTER}

This formula will place the four fields identified in the first line on a form with TAB keys in between each field

Below is a complete 2 line example of the Magnetic.txt file for the formula above:

1,2,4,6,9,2,1,5,6,5

q{TAB}r{TAB}s{TAB}t{ENTER}

If no second offset is required, then these fields are populated with zero and the unused lowercase field designators are not required

For example, 1,1,6,0,0,2,1,6,0,0

q{TAB}s{TAB}

Raw data can be extracted from each track by using the value 99 for the length.

If you want to extract the Raw Data from tracks 1 and 2:

1,0,99,0,0,2,0,99,0,0

q{TAB}s{TAB}

In Raw data mode for magnetic cards, if a track does not read, and ScanMsg is set to true then the output will be %NR? (track 1 no read) Or %ND? (track 1 no data)

If ScanMsg is set to false and a track does not read, the output will be just the sentinels,

E.G

Track1 no read /no data = %?

Track2 no read /no data = ;?

Track3 no read /no data = %

The Magnetic formula and parsing has been designed to handle cards on any of the 3 possible tracks. The Magnetic formula applies to cards that have 25 or less characters of information encoded on any track (Except old New Mexico ID/DL).

The following list of cards is used to identify Drivers License/ID cards, Credit Cards and AAA cards.

Track2 starts with	and a length of	Card Type
;000000	>28	old AZ state ID/DL
;06360	>28	old AK state ID/DL
;10	>36	Newfoundland ID/DL
;20	>36	Newfoundland ID/DL
;34	>28	AMEX Credit Card
;37	>28	AMEX Credit Card
;4	>31	VISA Credit Card
;4290	>36	AAA Membership Card
;4381	>36	AAA Membership Card
;4382	>36	AAA Membership Card
;51	>31	MasterCard Credit Card
;52	>31	MasterCard Credit Card
;53	>31	MasterCard Credit Card
;54	>31	MasterCard Credit Card
;55	>31	MasterCard Credit Card
;5490	>36	AAA Membership Card
;6006	>28	old CA state ID/DL
;6011	>31	Discover Credit Card
;610054	>28	Ontario Health ID Card
;6202	>36	AAA Membership Card
;6360	>28	United States ID/DL
;636005	>28	South Carolina ID/DL

NOTE: AAA membership cards fall into the Magnetic formula, Credit Cards fall into Credit Card parsing, and ID/DL cards are treated as Drivers Licenses.

Each of the above numbers is also tested for the presence of the '=' sign and a length, to help identify the card type. Cards that do not meet the test will be parsed using Magnetic formula. There is one exception and that is the South Carolina ID/DL, that card is not tested for the '=', only the length is tested.

The Location of the '=' sign for Credit cards use a value consistent with a 15 or 16 digit PAN.

CONTROL KEYS

The control keys are the non-printing keyboard strokes used in between the parsed field in the formula. The examples so far have included the Tab key {TAB} and the Enter key {ENTER}. The format of the control keys is an all-capital key word surrounded by curly braces (except for Pause). Each curly brace must be paired with another (left, right) or the formula will not work. Below is a list of the supported control keys that can be used in the formula. If multiple keystrokes are required, then an integer can be included inside the curly braces with a space between the key word, for example {TAB 2} will output 2 TAB characters. The values for repeating keys are 2 through 9.

{ALT x}	Alt plus key *
{CONTROL x}	Ctrl plus key *
{SHIFT x}	Shift plus key *
{CTLALT x}	Ctrl plus Alt plus key *
{CTLSHIFT X}	Ctrl plus Shift plus key *
{CTLALTS SHIFT}	Ctrl plus Alt plus Shift plus key *
{ALTSHIFT}	Alt plus Shift plus key *
{TAB}	Tab key
{ENTER}	Enter
{UP}	Up arrow
{DOWN}	Down arrow
{LEFT}	Left arrow
{RIGHT}	Right arrow
{PGUP}	Page up
{PGDN}	Page Down
{INS}	Insert
{DEL}	Delete (allows repeating)
{BACKSPACE}	Backspace (no repeating allowed)
{PAUSE x}	Pause in (x *100) Ms increments **

*Note: The control word is followed by a space and then the character to be acted upon, or the control word is followed by a hyphen and the hexadecimal value representing the key to be acted upon. See the Control Combination section for more details and a chart of hexadecimal key codes.

** Note: Pause command is case sensitive and must be followed by a space and then a numerical value from 1 thru 9, for example {Pause 5} will wait 500 Ms

CONTROL COMBINATION KEYS

The Control Combination keys support two formats for inputting the key to be acted upon.

The first format is to use the control word followed by a space and then either a lower case or upper case letter or number or symbol, one that is visible and can be represented by a single character. The character to be acted upon, will be reduced to the lower case version by the firmware, so if you need to use a capital letter for a specific keyboard short cut, you will need to use the shift key. Here is an example of this: {CONTROL f} will bring up the find dialog in MS Word, if you use {CONTROL F} it will still bring up the find dialog. If you want to bring up the font dialog, you need to use {CTLSHIFT F}.

The Excel spreadsheet app uses keyboard short cuts that demonstrate this type of format.

Combination	Definition
{CTLSHIFT ;}	Control key plus Shift plus the colon key, insert the time into a cell in Excel.
{CONTROL ;}	Control key plus the semi-colon key, insert the date into a cell in Excel.
{CONTROL p}	Control key plus lower case p, sends the Excel worksheet to the printer.
{CTLSHIFT p}	Control Key plus Shift plus P, brings up the format cell dialog in Excel.

The second format is to use a control word followed by a hyphen and then the two character hexadecimal code that represents the key to be acted upon. This format is useful for implementing the keys that do not have a corresponding visible character, like the ESCAPE key or the functions keys like F1.

Combination	Definition
{CTLSHIFT-28}	Control key plus Shift plus ENTER key, enter an array formula in Excel.
{SHIFT-4D}	Shift key plus END key, select to end of cell in Excel.
{CONTROL-3A}	Control key plus F1, Expand or collapse the ribbon on Excel.
{SHIFT-43}	Shift key plus F10 button, display right click menu in Excel

See appendix D for a list of supported hex codes.

INSERTING STATIC TEXT

To insert static text into the formula, simply enclose the text in square brackets []

You can insert a single character or a whole word. If you examine the formula from the *Driverslicense.txt* you can see the brackets are used to insert a comma into the output.

The Length of text is limited to 10 characters. If you exceed 10 characters within the brackets, an error message will be output if ScanMsg is enabled. If ScanMsg is disabled, you may not get any output from the scan.

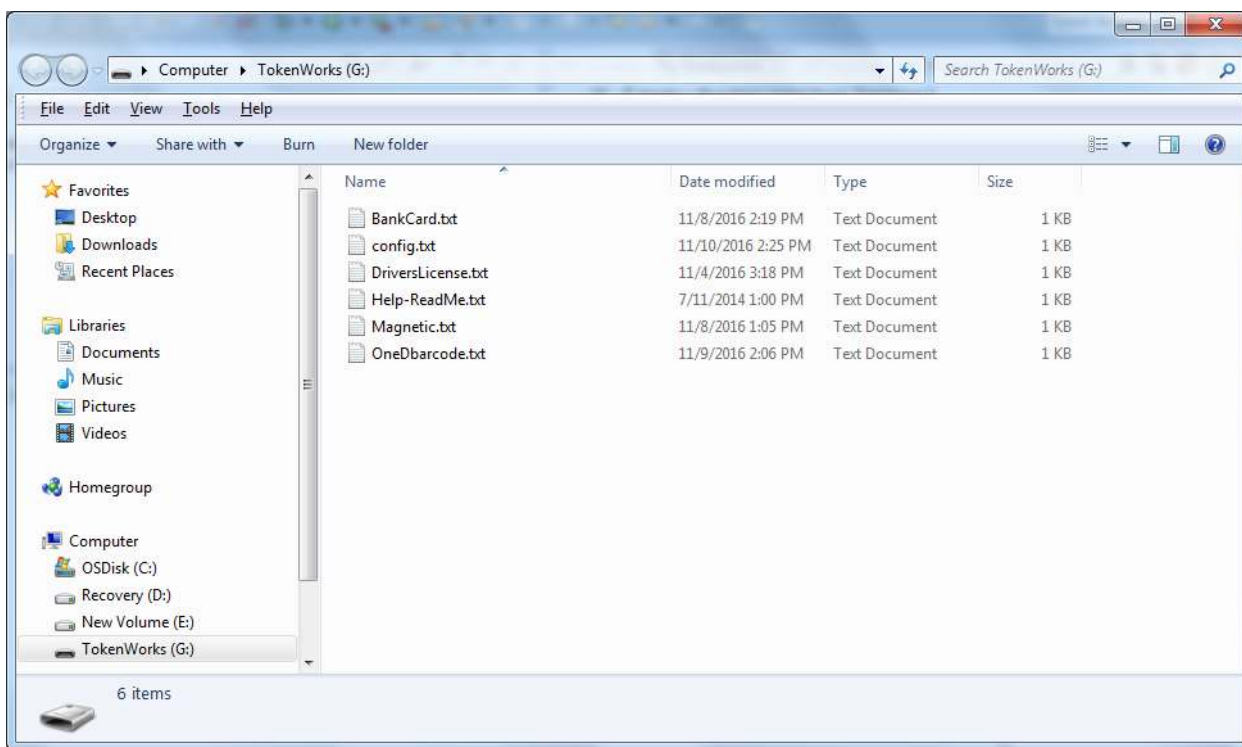
Example: Insert the word "Title=" into a formula using that field, plus the first and last name, followed by the return/enter key:

```
F{TAB}L{TAB}[Title=]T{ENTER}
```

FORMULA AND CONFIG FILE TESTING AND PROBLEM.TXT

The formula and configuration files are created by the IDWedgeKB when it is first powered up at the factory. User modifies these files to customize the behavior of IDWedgeKB. If a file is erased, the default version of the file will be re-created during next power cycle. Additionally, there are automatic test performed on these files during power up, to ensure the files adhere to some basic formatting rules. If an error is detected, a diagnostic file called: *Problem.txt* will be created and contain the error condition to be fixed, and the file that contains the error. Once the file's error(s) have been corrected, the *Problem.txt* file will be automatically deleted, if no other errors are detected, during the next power cycle.

The file system normally contains 6 files as seen in this screen shot:

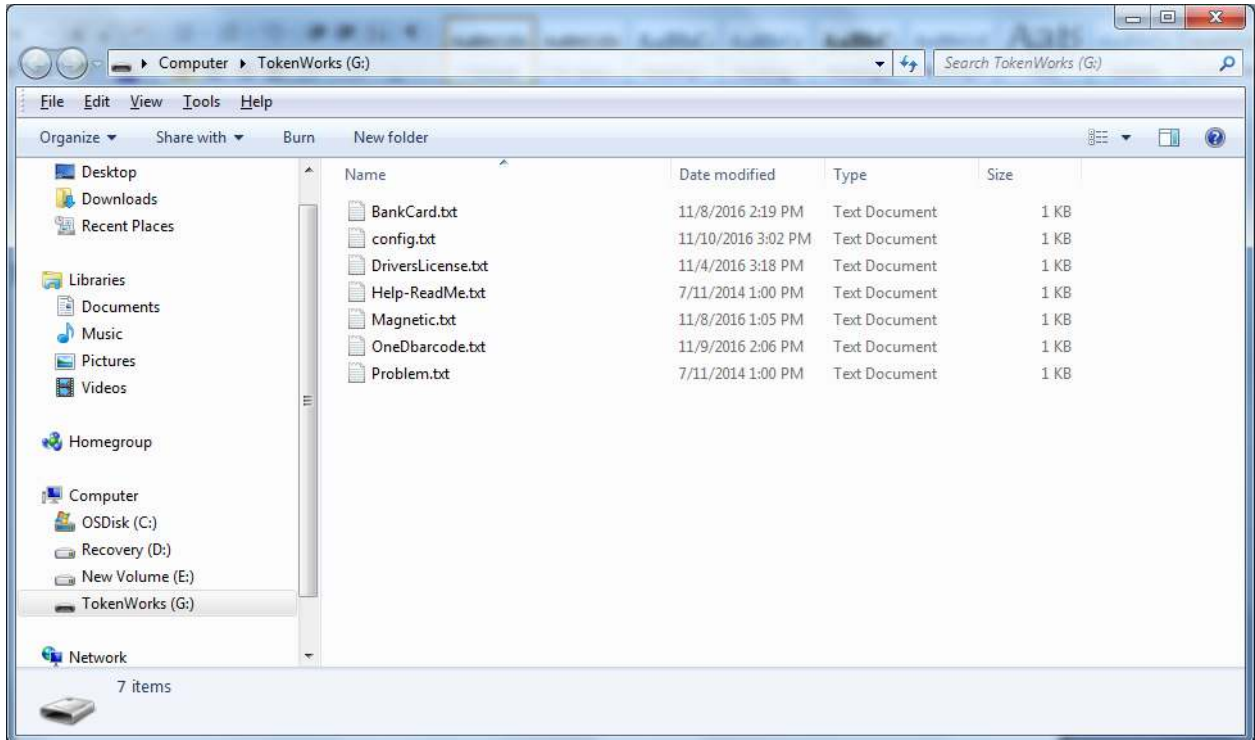


The following is a list of normal files when no problem is found:

BankCard.txt
Config.txt
DriversLicense.txt
Help-ReadMe.txt
Magnetic.txt
OneDBarcode.txt

If a problem is found:

There will be 7 items in the directory; the new item is called ***Problem.txt***.



An example of the contents of the file Problem.txt file will appear like this:

Reading CharDelay value of 0-9 fails in Config.txt

This file was created because a problem was found in the above file, it will be deleted when the problem is cleared.

To restore a default version of any file, just erase the file and reboot

For more details, See IDWedgeKB user manual at www.idscanner.com

TYPE OF TESTS AND EXAMPLE OF DIAGNOSTIC MESSAGES

The type of tests that are performed on the files depends on the contents, there are up to 4 different tests run on a file to determine if it meets minimum requirements.

CONFIG.TXT TEST AND MESSAGE

The configuration file contains four different tests. The first test looks for True/False values used to control application features, and a second test for numeric value for character delay, the third tests for the correct setting of the USB Keyboard and USB Serial Port variables, and the fourth test is for version control.

If the True/False test cannot find anything resembling true/false, then it will create the Problem.txt file and write the following message:

Reading True/False test fails in Config.txt

The character delay field is a single digit following the = sign, if the test cannot find a value that resembles the characters 0 thru 9, then it will create the Problem.txt file and write the following message:

Reading CharDelay value of 0-9 fails in Config.txt

If the True/False test cannot find that both of the mentioned variables are true or false, then it will create the Problem.txt file and write the following message:

Both USBKeyboard and USBSerialPort are either True or False in Config.txt

The version test looks at the version of the config.txt file and compares it to the internal firmware version, if they do not match, then it will create the Problem.txt file and write the following message:

Config.txt file version incorrect, erase file and reboot to reset to factory default

CARD FORMULA FILES TEST AND MESSAGE

There are three formula files: BankCard.txt, DriversLicense.txt, and OneDBarcode.txt. These files all have two tests performed on them. One test is to look for an even count of Curly Braces, used for control keys like {TAB}. The second test looks for an even count of Brackets, used for inserting text like [ID#]. If either of these two tests fails then the *Problem.txt* file will be created with one of the following messages:

Curly Brace { } Pair test fails in DriversLicense.txt

Or

Bracket Pair [] test fails in BankCard.txt

NOTE: These three formulas must consist of a single line of text.

MAGNETIC.TXT CARD FORMULA TEST AND MESSAGE

The Magnetic.txt file controls the parsing of loyalty cards or student ID cards with Magstripe encoding. It contains two lines, where the other formula files only contain 1 line. In the first line there must be two elements, there must be 9 commas delimiting the 10 offset and track designators, and there must be a carriage return. If either of these two tests fails, then the Problem.txt file will be created with one of the following messages:

Comma count not equal to 9 in Magnetic.txt

Or

Missing CR in Magnetic.txt

The second line of a magnetic.txt is like the other formula files, and will have one of the two possible error messages as seen in 4.1.2, with the file name set to Magnetic.txt.

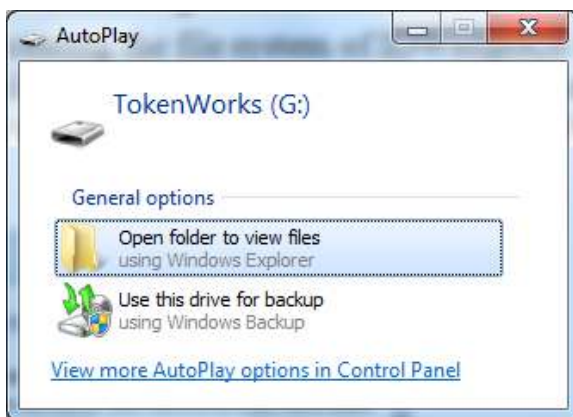
NOTE: These tests are run consecutively, and stop at the last error found, so it is possible to edit multiple files, and only be notified of a failure in one file. This would result in multiple boots to clear multiple errors. For this reason, it is suggested that only 1 file be modified per editing session until the user is familiar with the process.

ACCESSING THE FILE SYSTEM FOR EDITING

The process for accessing the file system of IDWedgeKB involves holding the side function button depressed, while plugging in the USB cable. After the initial install which may load some drivers, you will see a new drive dialog appear when you plug in the USB cable:



Select Continue without scanning, and you will see:



Open with Windows Explorer.

EDITING AND CLEARING PROBLEM.TXT

Once you have accessed the file system, you can use Windows Notepad app to edit the text files. Below is an example of a simple formula to print a two line address to a text file for DriverLicense.txt:

```
F{TAB}M{TAB}L{ENTER}A{TAB}C{TAB}S{TAB}Z{ENTER}
```

This example below shows an error in the number of braces (extra brace on end).

```
F{TAB}M{TAB}L{ENTER}}
```

This error formula will create the Problem.txt file with a message:

Curly Brace { } Pair test fails in DriversLicense.txt

Once you save a file, you must unplug the scanner from the USB port, and then plug it in for operational use (do not depress the side button). If an error was found, you will be notified the next time you access the file system. Some errors may be detected by the test scan not working; others may not impact behavior until after the unit is deployed. For this reason, we advise you to examine the file system to ensure that no error was detected before deploying.

ADDITIONAL INFORMATION AND EXAMPLES

Tips

Do NOT end the formulas with a tab as this will move cursor off last field and if there are no more fields on the form, then the keyboard may not be able to display because the cursor was moved off the form.

Example of the Pause command in a formula:

```
F{TAB}M{TAB}L{TAB}A{TAB}C{TAB}S{TAB}Z{PAUSE 5}
```

The above formula prints out a typical contact with name and address and after the last field, which is Zip code; there is a pause for 500 Milliseconds (1/2 second)

NOTE:

If you do not include an integer value preceded by a space the pause command will be ignored.

The pause command should be used sparingly and works best at the end of formulas, it was included to allow for time for the information to fill in the form, prior to launching the keyboard, however it could be used to control the cursor in between fields if the form requires a slower input.

CharDelay:

This variable is stored in the Config.txt file on the scanner memory. It controls the time to wait between sending out the next character. If the form needs extra time due to field verification or slow host processor, increase the value of the CharDelay. If you need to speed up the output of keystrokes, decrease the value of CharDelay. CharDelay =0 is the fastest rate you can achieve. The values 1 thru 9 represent 10 thru 90 Ms of delay time.

SERIAL PORT MODE

The Serial Port Mode sends the data in a fixed CSV string over the serial port to the host PC. The serial port operates at 115200 baud, 8 bit data, 1 stop bit, and parity = none. This mode is for application developers who will use the serial port to receive the parsed fields. The CSV output string for DL/ID cards and the CSV output string for credit cards are predefined, and provide all available fields from a card scan.

USB Virtual Com Port Driver with directions available for download at: <http://www.idscanner.us/idwedge-kb/idwedgekb-virtual-com-serial-port-inf-driver-signed/>

SERIAL PORT MODE *DL/ID CARD CSV STRING*

For Drivers License/ID cards the data will be output using a fixed formula string that represents each field, separated by a comma.

0x01 0x02 F,M,L,T,A,C,S,@,Z,D,E,B,X,H,I,U,P,O,V,?,&,#, 0x03 0x04

The outgoing CSV messages will be bounded by start and end characters:

Start: 0x01 0x02

End: 0x03 0x04

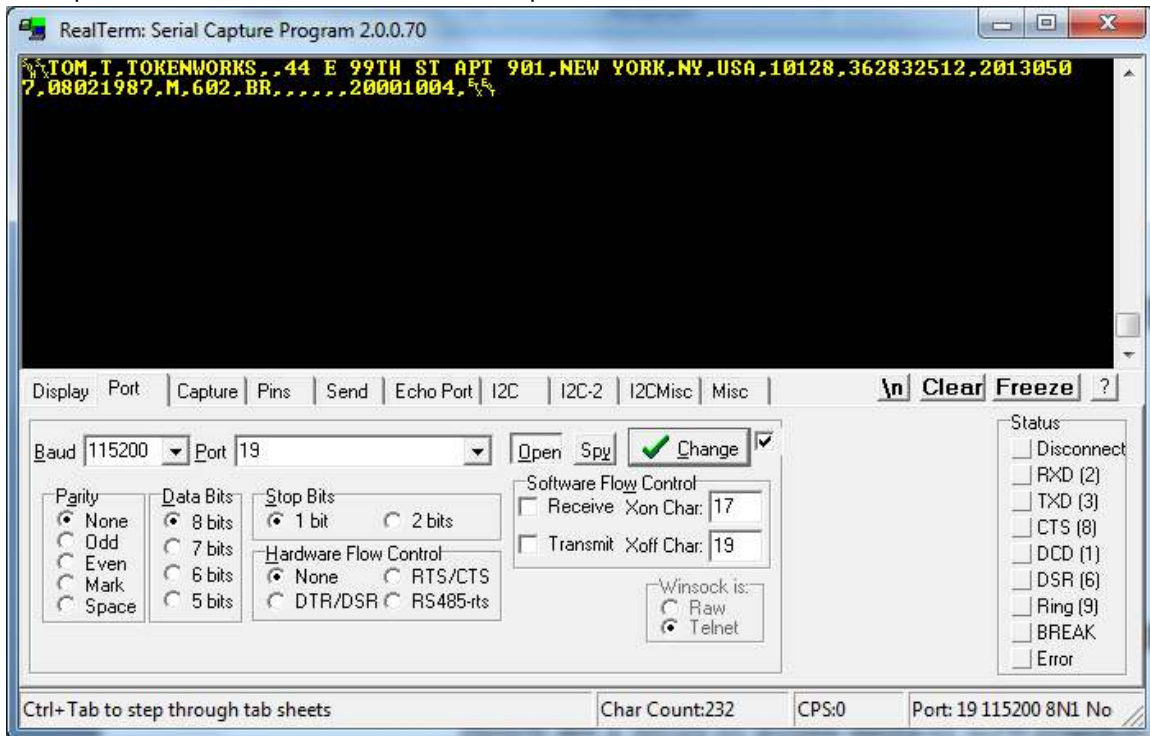
The letters in the above formula represent parsed fields from a DL/ID card.

F = First Name	Z = Zip code	U = Weight
M = Middle Name	D = ID #	P = Hair Color
L = Last Name	E = Expiration	O = Class
T = Title	B = DOB mmddyyyy	V = Endorsement
A = Address	X = Sex	? = Restriction
C = City	H = Height	& = DocIssueDate*
S = State	I = Eye color	# = 2 nd Address*
@ = Country*		

* Fields are only available on PDF417 2D Barcodes only.

** Rank is for CaC military Ids only (PDF417) and is output into the Title field. The Pay Grade uses the Class field (O), and the Base10 version of the ID number uses the Endorsement field(V).

Example of Drivers License Serial Mode Output:



SOH STX = 01 02 ETX EOT = 03 04

SERIAL PORT MODE *CREDIT CARD CSV STRING*

For credit cards the data will be output using a formula string that represents each field, separated by a comma.

0x01 0x02 *f,l,p,m,y* 0x03 0x04

f = First name
l = Last name
p = PAN (Card number)
m = EXP MM
y = EXP YY

Example of Bank Card Output:



SOH STX = 01 02 ETX EOT = 03 04

SERIAL PORT MODE *CREDIT CARD CCRAW=TRUE*



For credit cards, when CCRAW is set to True, the raw data for track1 and track2 will be used, instead of the normal credit card csv string

0x01 0x02 track1 track2 0x03 0x04

SERIAL PORT MODE *MAGNETIC CARDS CSV STRING*

The Magnetic cards will still require a formula, and that formula should be modified to include Commas to create the CSV string.

The *Magnetic.txt* stores the formula for Magnetic cards and must have two lines of data; the first line defines the track number, offset, and length associated with each field. The second line of the *Magnetic.txt* uses brackets to create the CSV string.

Example of first line of *Magnetic.txt*: 1,2,4,6,9,2,1,5,6,5

(Track 1, offset1 = 2, length1 = 4, offset2 = 6, length2 = 5, track 2, offset1 = 1, length1 = 5, offset2 = 6, length2 = 5).

Example of second line of *Magnetic.txt*:

q[,]r[,]s[,]t

Below is an example of the *Magnetic.txt* file for the formula above.

Example:

1,2,4,6,9,2,1,5,6,5

q[,]r[,]s[,]t

If no second offset is required, then these fields are populated with zero and the unused lowercase field designators are not required

For example, 1,1,6,0,0,2,1,6,0,0

q[,]s

If you want to extract the Raw Data from tracks 1 and 2:

1,0,99,0,0,2,0,99,0,0

q[,]s

Below is an example of a *Magnetic.txt* file and the output:

1,2,4,6,9,2,1,5,6,5

q[]r[]s[]t[]

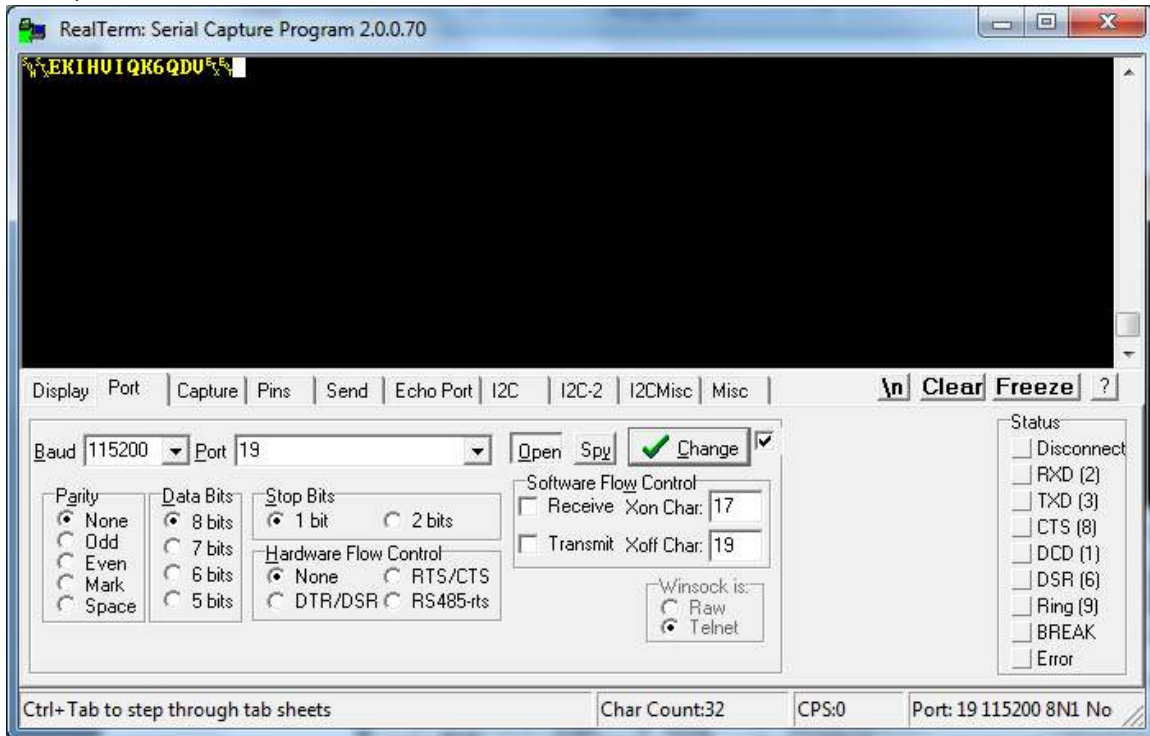


The formula grabs two fields from track1, and two fields from track2. The first field is 4 bytes, starting at offset of 2. The second field is a 9 bytes starting at offset 6. The third field is 5 bytes starting at offset of 2 and the forth field is also 5 bytes with an offset of 6.

EXAMPLES OF SERIAL PORT MODE OUTPUT 1D BARCODE MESSAGES

For 1D barcodes – if enabled, the output will be the start / end characters plus the 1D barcode.

Example of 1DBarcode:



SOH, STX = 01,02 ETX,EOT = 03,04

DOCUMENT CHANGE LOG

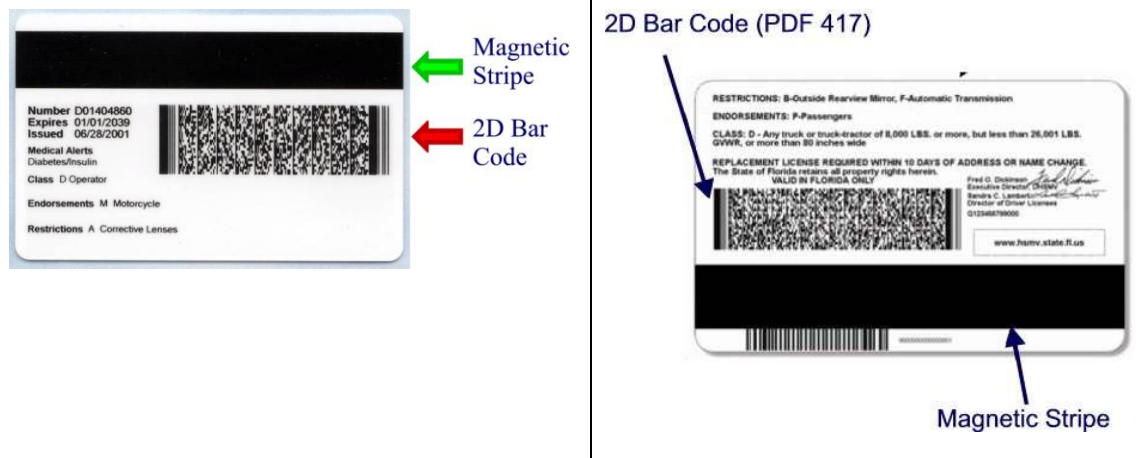
Version	Description of Change	Author	Date
1.0	Initial	PC	10272016
1.01	Format Changes	CWC	11012016
1.02	Add file system test feature	PC	11102016
1.03	Review latest firmware, create open issues at end of document, and highlight in yellow in doc	PC	12142016
1.04	Resolve open issues	PC	12262016
1.05	Correct file name error Config.xml to Config.txt	PC	02152017
1.0.6	Add SPP profile	PC	06202017
1.0.7	Major Cleanup – renamed SPP – added inf file for serial port -	CC	08082017
1.08	Review and finalize	PC	08092017
1.09	Removed Appendix Z – Removed numbering from chapter 4 and parts of 3	CC	08092017
1.10	Upgrade firmware to 2.4, CaC card now uses Class field for pay grade and Endorse field for base10 ID#	PC	08232017
1.11	Updated led flash signal chart	CC	08232017
1.12	Correct Drivers License field order for SPP	PC	01052018
1.13	Add the {CTLSHIFT} control word for ver. 2.7	PC	08162018
1.14	Add all control word combination ver. 2.8	PC	08212018
1.15	Fix bug that was filtering dash/underscore char	PC	10032018
1.16	Fix bug that stopped 1D barcode code39 output	PC	03012019
1.17	Add Gender X to AAMVA barcodes, add new Magstripe for MN, document Dob/Exp format	PC	03202019
1.18	Add CCRaw for use in Serial Port Mode	PC	10162019
1.19	Parsing update for new FL magstripe v3.3	PC	04152020
1.20	Add Real ID field for barcodes, Formula = G v3.4	PC	08102020
1.2.1	Update Mag parsing for CA DL# AlphaNum v3.5	PC	03092022

LICENSE DATA STRIPE TECHNOLOGY

The ID Scanner Reads 2D Bar Codes and Magnetic Stripes. This section explains how to recognize which is on the driver's license. For latest details on which state has which technology – visit <https://www.tokenworks.com/coverage-chart/>

If a license has both, we suggest you scan the 2D Bar code first and the magnetic stripe to verify the information on the screen is the same from both scans.

Note: It is advised NOT to enable 1D barcode when scanning drivers licenses because some states 1D barcodes depending on how ID is inserted in scanner, will decode instead of 2D barcode.



MILITARY IDS

To Scan Military IDs, SCAN the **2D Bar Code**.

DO NOT SCAN THE 1D BAR CODE OR MAGNETIC STRIPE

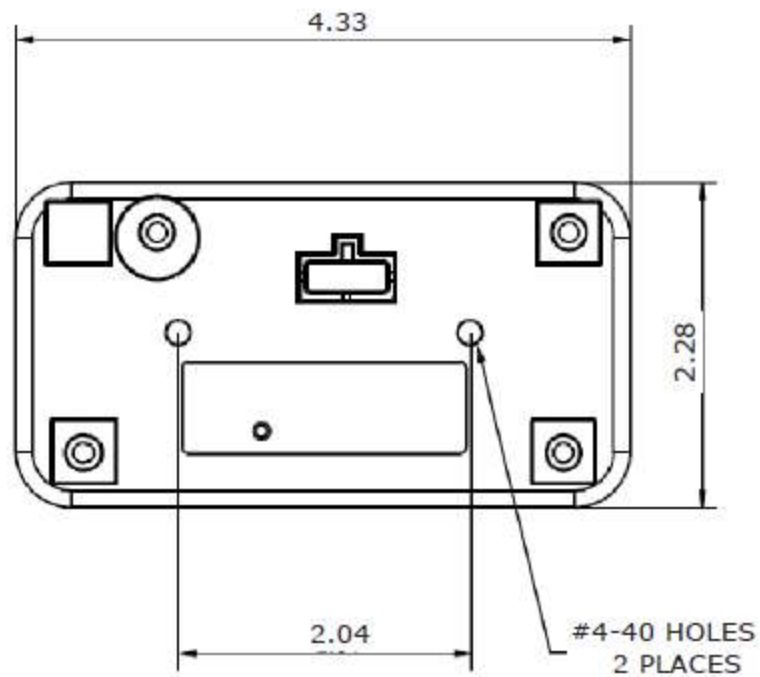
Do Scan 2D Bar Code

Do NOT Swipe Magnetic Stripe or 1D Bar Code

Specifications

Decoder Two Dimensional	PDF417
Linear Barcodes	Code 39 & Code 128
Card Operating Mode	Hands-free, Manual Insertion of ID card or Magnetic Stripe Swipe action
Interface	- RS-232C Serial Port - USB 2.0 Full speed (w/FTDI drivers, Windows certified)
Power Consumptions	4 mA @ 5VDC - Standby; 250 mA Maximum
Dimensions	2.13" H x 2.28" W x 4.33" D
Weight	0.5 Pound
Card Size	ISO/IEC-7811, ID-1 Standard Size 3.370" x 2.125"
Housing	ABS Plastic in Black
Operating Temperature	32 to 122° F (0 to 50° C)
Humidity	10% to 90%, non-condensing

Mounting picture. (Inch)

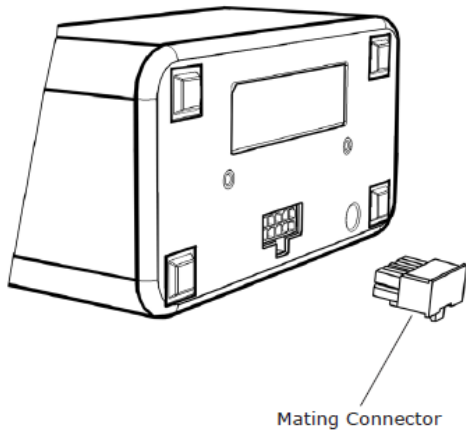


APPENDIX C – OPTIONAL KIOSK CABLE

An optional Kiosk connector and cable allow fixed mounted installations for Kiosks. This connector and cable are used in place of the standard side USB cable which has RJ50 connector

Installing the Kiosk Cable

Simply connect the kiosk male connector on the interface cable to kiosk female connector built into the Models M260 as per the illustration below.



CN8436 Kiosk Cable

APPENDIX D – HEXADECIMAL KEY CODE CHART

HEX value	Key
04	a and A
05	b and B
06	c and C
07	d and D
08	e and E
09	f and F
0A	g and G
0B	h and H
0C	i and I
0D	j and J
0E	k and K
0F	l and L
10	m and M
11	n and N
12	o and O
13	p and P

14	q and Q
15	r and R
16	s and S
17	t and T
18	u and U
19	v and V
1A	w and W
1B	x and X
1C	y and Y
1D	z and Z
1E	1 and !
1F	2 and @
20	3 and #
21	4 and \$
22	5 and %
23	6 and ^
24	7 and &
25	8 and *

26	9 and (
27	0 and)
28	Return (ENTER)
29	ESCAPE
2A	Backspace
2B	Tab
2C	Spacebar
2D	- and _
2E	= and +
2F	[and {
30] and }
31	\ and
32	Non –US # or ~
33	; and :
34	' and “
35	` and ~
36	, and <
37	. and >

38	/ and ?
39	CapsLock
3A	F1
3B	F2
3C	F3
3D	F4
3E	F5
3F	F6
40	F7
41	F8
42	F9
43	F10
44	F11
45	F12
46	PrintScreen
47	ScrollLock
48	Pause
49	Insert
4A	Home

4B	PageUp
4C	Delete Forward
4D	End
4E	PageDown
4F	RightArrow
50	LeftArrow
51	DownArrow
52	UpArrow