

---

# **IDWedgeBT™ Users Guide**

## **USB Keyboard Mode**

### **Version 2.7 Software**



# Copyright

Copyright 2002 - 2020 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](http://www.TokenWorks.com) (Company Web Site)

[www.CardVisor.com](http://www.CardVisor.com), [www.IDVisor.com](http://www.IDVisor.com), [www.IDScanner.com](http://www.IDScanner.com) (Product Web site)

TokenWorks<sup>®</sup>, CardVisor<sup>®</sup>, CardTool<sup>®</sup>, IDVisor<sup>®</sup> and IDWedge<sup>®</sup> are registered trademarks of TokenWorks, Inc. IDWedgeBT Keyboard is a TokenWorks Inc trademark.

Version	Description of Change	Author	Date
1.0	Initial	PC	01182013
1.1	Add improved Magnetic functionality	PC	03082013
1.2	Add Doc Issue Date	PC	03092013
1.3	Add FuncKey F1-F12	PC	03112013
1.4	Improve output speeds, allow longer formula, add USB connection light and FuncKey lights	PC	04152013
1.5	Parse DOB, EXP into mm dd yyyy fields Also add raw data output for Credit cards	PC	04222013
1.6	Add Raw data form Magnetic cards and fix Bug with Pause cmd	PC	04292013
1.7	Change power up initialization	PC	05312013
1.8	Fix IL parsing, add 2 <sup>nd</sup> Addr, Country, Rank and 1 <sup>st</sup> initial only, code 39 1D barcode support	PC	08262013
2.1	No firmware changes, make s/w rev = doc rev	PC	03312014
2.2	Parsing update for MN PDF417 barcode	PC	10232015
2.3	Add DisableMagDL_CC feature	PC	06132016
2.4	Fix USAA Visa card length issue, y2k dob	PC	01262017
2.5	Fix bug with {LEFT} and Country to PDF	PC	05172018
2.6	Add Gender X, new MN Magstripe	PC	03192019
2.7	Add Real ID to PDF, new FL Magstripe	PC	08112020

<b>Copyright</b> .....	<b>2</b>
<b>1 Introduction</b> .....	<b>5</b>
1.1 Document overview.....	5
1.2 Operational overview.....	5
1.2.1 Power .....	6
1.2.2 Battery .....	6
1.2.3 Function Button .....	6
1.2.4 Connection Status .....	6
<b>2 Configuration file</b> .....	<b>7</b>
2.1 <i>Config.xml</i> .....	7
2.1.1 Version.....	8
2.1.2 FuncKey.....	8
2.1.3 CreditCard.....	8
2.1.4 1DBarcode.....	8
2.1.5 MagTrackCard.....	8
2.1.6 MidInitial .....	8
2.1.7 FirstInitial.....	8
2.1.8 ScanMsg.....	8
2.1.9 5DigitZip.....	8
2.1.10 DobYYOnly.....	9
2.1.11 ExpYYOnly .....	9
2.1.12 CharDelay .....	9
2.1.13 CtrlDelay.....	9
2.1.14 DisableMagDL_CC .....	9
2.2 <i>Reading/Writing Config.xml</i> .....	9
<b>3 Formula Files</b> .....	<b>10</b>
3.1 <i>Driverslicense.txt</i> .....	11
3.2 <i>CreditCard.txt</i> .....	12
3.3 <i>OneDbar.txt</i> .....	12
3.4 <i>Magnetic.txt</i> .....	12
3.5 <i>Control keys</i> .....	15
3.6 <i>Inserting text</i> .....	15
<b>4 Additional Information and Examples</b> .....	<b>16</b>
<b>Appendix 1 – Power on BIT (Built in Test) Description</b> .....	<b>17</b>
<b>Appendix 2 – LED interpretation</b> .....	<b>17</b>



# 1 Introduction

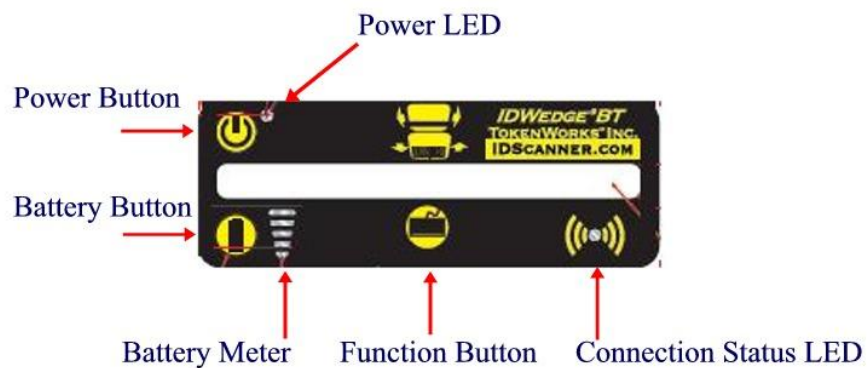
The IDWedgeBT Keyboard is a barcode/magnetic card scanner that supports a USB connection to a host computer/tablet and functions as a HID/Keyboard device. Cards are scanned and parsed into fields, and these fields are sent as keyboard strokes to the host. The sequence of fields sent is controlled by formulas stored on the IDWedgeBT Keyboard SD memory card.

## 1.1 Document overview

This document will describe the basic functionality of the IDWedgeBT Keyboard and how to configure the application parameters and formulas used to control the output of data from the scanner.

## 1.2 Operational overview

The diagram below shows the top view of the IDWedgeBT Keyboard scanner and the location of the LED indicators and button switches used during the operation of the device.



### **1.2.1 Power**

The Power Button switches the battery power on/off to the unit. Power can be applied by connecting the 12VDC power supply, which charges the internal battery. The USB cable can also supply power, but does not charge the battery. If either of these two cables is connected the device powers on and ignores the state of the Power Button. When the device is powered on, the device will triple beep, and perform a Built in Test. After approximately 5 seconds the unit is ready for Scanning. See Appendix 1 for a description of the Built in Test (BIT)

### **1.2.2 Battery**

The Battery button displays the state of the internal battery, press it once and view the state of the battery on the 5 segment LED. The Battery button serves a dual purpose of providing access to the SD card. To access the SD card, with the power off, press and hold the Battery button and then power on the device. Continue to hold the Battery button for 3-4 seconds, and then release it. When the USB cable is connected to a PC, the IDWedgeBT Keyboard will now appear as an USB memory device, and the Connection Status LED will stay lit.

NOTE: The USB cable supplied with the Scanner supplies operational power. The preferred method of accessing the SD card is to hold the battery button down, and plug in the USB cable to the PC and wait until the PC recognizes the SD card.

### **1.2.3 Function Button**

The Function button allows the user to send one of the function keys from a standard PC keyboard F1 through F12. Pressing the button also lights the lower 3 LED segments of the 5 bar display, to identify the IDWedgeBT USB keyboard application.

### **1.2.4 Connection Status**

This LED will stay on solid when the device is in SD card mode and will extinguish after the next power cycle. The SD card mode allows the device to be recognized by a PC as an external disk drive, via USB cable.

## 2 Configuration file

The IDWedgeBT Keyboard maintains configuration file called *Config.xml*. This file stores the application parameters. If the file gets erased, it will be created automatically using default settings on power on. If the file contains editing errors or is corrupted and cannot be opened, then a file called *ConfigError.txt* will be written to the SD card. Once the errors have been corrected and the file can be read, the *ConfigError.txt* file will be automatically erased.

### 2.1 Config.xml

Below is an example of the contents of the default *Config.xml* file.

```
<add key=" Version" value="2.7" />
<!--version string, do not modify. Used to control updates to Config.xml -->
<add key=" FuncKey" value="255" />
<!-- Keystroke sent when keyboard button is pressed -->
<add key=" CreditCard" value="True" />
<!-- Controls how credit card data is parsed output - creditcard.txt -->
<add key=" 1Dbarcode" value="True" />
<!-- Controls how 1Dbarcodes output in HID mode - OneDbar.txt -->
<add key=" MagTrackCard" value="True" />
<!-- Controls how mag stripes are parsed output in HID mode - Magnetic.txt -->
<add key=" MidInitial" value="False" />
<!--Only send first digit of Middle name field -->
<add key=" FirstInitial" value="False" />
<!--Only send first digit of First name field -->
<add key=" ScanMsg" value="True" />
<!--Send out Scan again message if card does not read -->
<add key=" 5DigitZip" value="False" />
<!--Only send first 5 digits of Zip Code field -->
<add key=" DobYYOnly" value="True" />
<!--Only send last2 digits of YYYY field -->
<add key=" ExpYYOnly" value="True" />
<!--Only send last 2 digits of YYYY field -->
<add key=" CharDelay" value="0" />
<!--inter-character delay X*10Ms (range is 0-19) -->
<add key=" CtrlDelay" value="0" />
<!--Control-character delay X*20Ms (range is 0-19) -->
<add key=" DisableMagDL_CC" value="False" />
<!--Control-character delay X*20Ms (range is 0-19) -->
```

### **2.1.1 Version**

This is the firmware version of the IDWedgeBT Keyboard application. E.g. 2.7

### **2.1.2 FuncKey**

FuncKey defines the Function keys F1-F12 that will be sent when the FuncKey button is pressed. The range for this field is 58-69. A value of 58 = F1, 59 = F2, etc. The default value is 255, which means, “Do nothing” if a value is entered outside of 58-69, then the default value/action will occur.

### **2.1.3 CreditCard**

CreditCard controls the output of Credit Card scans

### **2.1.4 1DBarcode.**

1DBarcode controls the output of code128 and code 39 1D barcode scans

### **2.1.5 MagTrackCard**

MagTrackCard controls the output of Magstripe cards like Student Ids, AAA cards, loyalty cards, etc. To have all cards use this formula, set the DisableMagDL\_CC feature to True.

### **2.1.6 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.

### **2.1.7 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.

### **2.1.8 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 may cause extra operator effort when attempting to fill out forms.

### **2.1.9 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.



### **2.1.10 DobYYOnly**

The DobYYOnly variable only sends out the last two digits of the YYYY DOB field

### **2.1.11 ExpYYOnly**

The ExpYYOnly variable only sends out the last two digits of the YYYY EXP field

### **2.1.12 CharDelay**

The CharDelay controls the amount of time to wait between the transmissions of characters, the range is 0-20 and this number is multiplied by 10Ms. Any value outside this range will default to 1 ms.

### **2.1.13 CtrlDelay**

The CtrlDelay controls the amount of time to wait between the transmissions of Control characters, the range is 0-20 and this number is multiplied by 20Ms. Any value outside this range will default to 1 ms.

### **2.1.14 DisableMagDL\_CC**

The DisableMagDL\_CC feature works in tandem with the MagTrackCard feature, when both are set to True; all magstripe cards will use the magnetic formula. The state of the CreditCard feature is ignored when this feature is set to true.

## **2.2 Reading/Writing Config.xml**

To Access the Config.xml file, you must boot the IDWedgeBT Keyboard while holding the battery button as described in section 1.2.2. Once you have connected a USB cable and can explore the contents of the SD memory card, you will be able to read and write the *Config.xml* file. You can use any text editor to change items in the file and save them to the SD card. To ensure you have not made any errors, open up the Config.xml file using an Internet browser, if a browser cannot open the file, then there probably are editing mistakes that must be corrected before the IDWedgeBT Keyboard can be operational. The changes made the Config.xml will take effect after the next power cycle.

### 3 Formula Files

The Formula used to send fields to the host is stored in a file on the SD card. To Access the Config.xml file, you must boot the IDWedgeBT Keyboard while holding the battery 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, *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. The entire formula must not exceed 500 characters in length.

### 3.1 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 (YYYYMMDD)

**E** - EXP (YYYYMMDD)

**D** - License number

**H** - Height

**I** - Eye color

**O** - Class

**P** - Hair

**U** - Weight

**V** - Endorsements

**X** - Sex

**N** - Scanner S/N

**?** - Restrictions

**&** - Document Issue Date \*

**R** - Rank (CaC only)

**@** - Country (USA or CAN) \*

**#** - 2nd Address field \*

**G** - Real ID \*

**Q** - DOB\_mm

**W** - DOB\_dd

**Y** - DOB\_yy (yyyy format via config.xml variable)

**J** - EXP\_mm

**K** - EXP\_dd

**%** - EXP\_yy (yyyy format via config.xml variable)

Example:

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

Note:

Users can use the Q, W, Y formula field to create any format they desire for the DOB field, and J, K, % for the EXP field

\* (Available on PDF417 2D Barcodes only)

### 3.2 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.**

### 3.3 OneDbar.txt

There is only 1 field on a 1D barcode and that is represented with a lowercase **w**

Example:

```
w{ENTER}
```

### 3.4 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.xml file, you must boot the IDWedgeBT Keyboard while holding the battery button as described in section 1.2.2

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 as 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

The *Magnetic.txt* must have two lines of data; 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.

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).

The second line of the *Magnetic.txt* uses the lowercase field designators, along with the control words in curly braces, to place the fields on the form

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

Example:

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}

In version 1.6 and later, 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}

**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, then 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 are less than 28 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 base on the track2 data:

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	>=28	VISA Credit Card
;4290	>36	AAA Membership Card
;4381	>36	AAA Membership Card
;4382	>36	AAA Membership Card
;51	>=28	MasterCard Credit Card
;52	>=28	MasterCard Credit Card
;53	>=28	MasterCard Credit Card
;54	>=28	MasterCard Credit Card
;55	>=28	MasterCard Credit Card
;5490	>36	AAA Membership Card
;6006	>28	old CA state ID/DL
;6011	>=28	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 track two is also tested for the presence of the '=' sign to help identify the card type. If the '=' sign or Field Separator is found at location 17, it is probably a credit card. If a card has no '=' sign, and begins with ;6360 it is probably a Drivers license. Cards that do not meet these tests will be parsed using Magnetic formula.

If MagTrackCard and DisableMagDL\_CC are both set to True, then all magstripe cards will use the Magnetic formula. (See 2.1.14)

### 3.5 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 *
{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
{BACKSPACE}	Backspace
{Pause x}	Pause in (x *100) Ms Increments **

\*Note: Alt, Ctrl, Shift are used in combination with a single case insensitive character in the range of 0-9 or a-z or A-Z. These commands do not support repeating, so a command like {SHIFT 8} will produce the \* character, likewise {CONTROL 1} will be interpreted as the Ctrl key plus the 1 key (Ctrl +1).

\*\* Note: Pause command is case sensitive and must be followed by a space and then a numerical value from 1 –9

### 3.6 Inserting 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.

## 4 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 type Pause with all capitals it will be ignored, if you type in all lower case it will be ignored. If you do not include an integer value preceded by a space it 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 and CtrlDelay:

These two variables are stored in the Config.xml file on the SD card. They control the time to wait between sending out the next character (CharDelay) or next control character (CtrlDelay). If the form needs extra time due to field verification or slow host processor, increase the value of the CharDelay and or CtrlDelay. If you need to speed up the output of keystrokes, decrease the value of CharDelay and or CtrlDelay.

### Limitations:

For Code39 and Code128 (1D Barcodes); the card must have at least 8 characters encoded or the scanner will not read the card.



## **Appendix 1 – Power on BIT (Built in Test) Description**

On power up, the device will perform a self test and display the results on the 5 bar LED

### BIT Sequence

LED test, turn on/off each LED one at a time, then all 5 on/off

LED 1 - SD card Test, verify file create/read/erase

LED 2 - Test ESEEK Communication, verify send/receive hardware signals

LED 3 - Read Device.txt and verify BT revision string and ESEEK S/N string

LED 4, 5 - Cumulative Pass/Fail - if LEDs 1-4 are lit, then turn on LEDs 4, 5

Note: If the Config.xml file is erased, the Scanner will triple beep on the next power cycle, this is confirmation that the scanner has received the initialization command and has been configured to read barcodes and magstripe card. During normal operation there will be on triple beep heard on power up.

## **Appendix 2 – LED interpretation**

When the Function button is pressed the bottom 3 LEDs (LED1-3) will light up to identify the IDWedgeBT Keyboard application. The IDWedgeBT can run other applications, the HID application will only light up the first LED, and the SPP application will light up LED1 and 2. This is helpful for all users to identify what application is running on the device.

The Connection Status LED will only light when the device is in “SD Card Access” mode, see section 1.2.2. The LED will stay light until the device has been turn off/on.