MAGTEK®
SECURITY FROM THE INSIDE

MICRSafe

Check Reader and Magnetic Stripe Reader
Technical Reference Manual

April 2019

Document Number:
D99875466-33

REGISTERED TO ISO 9001:2015

Table 0-1 - Revisions

Rev Number	Date	Notes
1.01	Mar 3, 2010	Initial Release
2.01	Aug 27, 2010	Updated images; change AC to DC; updated command information
3.01	Oct 20, 2010	Noted configuration 8.3; In troubleshooting guide changed18 “Return MICRSAFE to MagTek” to “Return MICRSAFE for service”; Removed 08 from troubleshooting guide
32	Mar 20, 2018	Add section **About Encrypted** Check Data to include information about how encrypted data comes in from check scan: How to interpret, how to decrypt
33	Apr 25, 2019	Modernize format; Clarify that before using **Command 0x00 and Command 0x01 - Get Property and Set Property**, the host must first save any previously issued legacy setting changes; Misc. clarifications and corrections.

LIMITED WARRANTY

MagTek warrants that the products sold pursuant to this Agreement will perform in accordance with MagTek's published specifications. This warranty shall be provided only for a period of one year from the date of the shipment of the product from MagTek (the "Warranty Period"). This warranty shall apply only to the "Buyer" (the original purchaser, unless that entity resells the product as authorized by MagTek, in which event this warranty shall apply only to the first repurchaser).

During the Warranty Period, should this product fail to conform to MagTek's specifications, MagTek will, at its option, repair or replace this product at no additional charge except as set forth below. Repair parts and replacement products will be furnished on an exchange basis and will be either reconditioned or new. All replaced parts and products become the property of MagTek. This limited warranty does not include service to repair damage to the product resulting from accident, disaster, unreasonable use, misuse, abuse, negligence, or modification of the product not authorized by MagTek. MagTek reserves the right to examine the alleged defective goods to determine whether the warranty is applicable.

Without limiting the generality of the foregoing, MagTek specifically disclaims any liability or warranty for goods resold in other than MagTek's original packages, and for goods modified, altered, or treated without authorization by MagTek.

Service may be obtained by delivering the product during the warranty period to MagTek (1710 Apollo Court, Seal Beach, CA 90740). If this product is delivered by mail or by an equivalent shipping carrier, the customer agrees to insure the product or assume the risk of loss or damage in transit, to prepay shipping charges to the warranty service location, and to use the original shipping container or equivalent. MagTek will return the product, prepaid, via a three (3) day shipping service. A Return Material Authorization ("RMA") number must accompany all returns. Buyers may obtain an RMA number by contacting MagTek Support Services at (888) 624-8350.

EACH BUYER UNDERSTANDS THAT THIS MAGTEK PRODUCT IS OFFERED AS-IS. MAGTEK MAKES NO OTHER WARRANTY, EXPRESS OR IMPLIED, AND MAGTEK DISCLAIMS ANY WARRANTY OF ANY OTHER KIND, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IF THIS PRODUCT DOES NOT CONFORM TO MAGTEK'S SPECIFICATIONS, THE SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. MAGTEK'S LIABILITY, IF ANY, SHALL IN NO EVENT EXCEED THE TOTAL AMOUNT PAID TO MAGTEK UNDER THIS AGREEMENT. IN NO EVENT WILL MAGTEK BE LIABLE TO THE BUYER FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF, OR INABILITY TO USE, SUCH PRODUCT, EVEN IF MAGTEK HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

LIMITATION ON LIABILITY

EXCEPT AS PROVIDED IN THE SECTIONS RELATING TO MAGTEK'S LIMITED WARRANTY, MAGTEK'S LIABILITY UNDER THIS AGREEMENT IS LIMITED TO THE CONTRACT PRICE OF THIS PRODUCT.

MAGTEK MAKES NO OTHER WARRANTIES WITH RESPECT TO THE PRODUCT, EXPRESSED OR IMPLIED, EXCEPT AS MAY BE STATED IN THIS AGREEMENT, AND MAGTEK DISCLAIMS ANY IMPLIED WARRANTY, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

MAGTEK SHALL NOT BE LIABLE FOR CONTINGENT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES TO PERSONS OR PROPERTY. MAGTEK FURTHER LIMITS ITS LIABILITY OF ANY KIND WITH RESPECT TO THE PRODUCT, INCLUDING NEGLIGENCE ON ITS PART, TO THE CONTRACT PRICE FOR THE GOODS.

MAGTEK'S SOLE LIABILITY AND BUYER'S EXCLUSIVE REMEDIES ARE STATED IN THIS SECTION AND IN THE SECTION RELATING TO MAGTEK'S LIMITED WARRANTY.

FCC INFORMATION

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.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

CANADIAN DOC STATEMENT

This digital apparatus does not exceed the Class B limits for radio noise from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conformé à la norme NMB-003 du Canada.

CE STANDARDS

Testing for compliance with CE requirements was performed by an independent laboratory. The unit under test was found compliant with standards established for Class B devices.

UL/CSA

This product is recognized per ***UL 60950-1, 2nd Edition, 2011-12-19*** (Information Technology Equipment - Safety - Part 1: General Requirements), ***CSA C22.2 No. 60950-1-07, 2nd Edition, 2011-12*** (Information Technology Equipment - Safety - Part 1: General Requirements).

ROHS STATEMENT

When ordered as RoHS compliant, this product meets the Electrical and Electronic Equipment (EEE) Reduction of Hazardous Substances (RoHS) European Directive 2002/95/EC. The marking is clearly recognizable, either as written words like “Pb-free,” “lead-free,” or as another clear symbol (Pb).

SOFTWARE LICENSE AGREEMENT

IMPORTANT: YOU SHOULD CAREFULLY READ ALL THE TERMS, CONDITIONS AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE INSTALLING THE SOFTWARE PACKAGE. YOUR INSTALLATION OF THE SOFTWARE PACKAGE PRESUMES YOUR ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE PACKAGE AND ASSOCIATED DOCUMENTATION TO THE ADDRESS ON THE FRONT PAGE OF THIS DOCUMENT, ATTENTION: CUSTOMER SUPPORT.

TERMS, CONDITIONS, AND RESTRICTIONS

MagTek, Incorporated (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software."

LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with MagTek products. LICENSEE MAY NOT COPY, MODIFY, OR TRANSFER THE SOFTWARE IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt to reverse engineer the Software. Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software or license to the Software to another party without the prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted. Licensee may not copy the Software except for archival purposes or to load for execution purposes. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded are free from defects in material or workmanship under normal use.

THE SOFTWARE IS PROVIDED AS IS. LICENSOR MAKES NO OTHER WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Because of the diversity of conditions and PC hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE, THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s) if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void, or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the laws of the State of California and shall inure to the benefit of MagTek, Incorporated, its successors or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS, AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL VERBAL AND WRITTEN COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO MAGTEK, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE ADDRESS LISTED IN THIS DOCUMENT, OR E-MAILED TO SUPPORT@MAGTEK.COM.

Table of Contents

1 Introduction

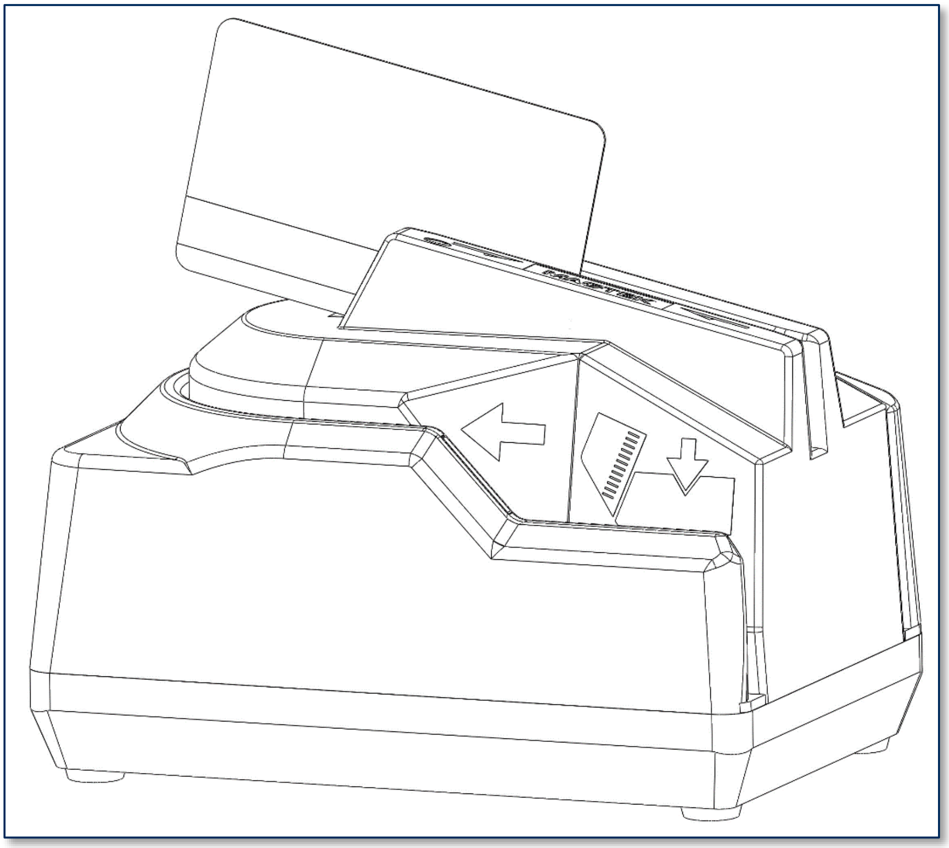


Figure 1-1 - MICRSafe with 3-Track MSR

MICRSafe with 3-Track MSR is both a MICR (Magnetic Ink Character Recognition) check reader and an MSR (magnetic stripe reader).

MICRSafe, in a typical application, reads the magnetic data encoded on the bottom of checks or on magnetic stripe cards and transmits this data to a host device. The host device then uses a specific authorization or verification process to validate a business transaction.

Using MICRSafe improves accuracy and speed because there is no manual data entry; therefore there are no keying errors or unwanted delays.

MICRSafe has three interface capabilities. First, MICRSafe can communicate with the host system using a standard USB interface. The driver emulates a serial port on the host PC. All data is transmitted as ASCII characters (See **Appendix E ASCII Codes**). This is MICRSafe VCOM device.

Second, MICRSafe can emulate a USB keyboard. This device is compatible with PCs or hosts that support USB keyboards. The Reader emulates a USB Human Interface Device (HID) United States keyboard or optionally all international keyboards using ALT ASCII code keypad key combinations or customizable key maps. This allows host applications designed to acquire card data from keyboard input to seamlessly acquire the card data from the reader. This is MICRSafe HID device.

⚠ CAUTION

If another keyboard is connected to the same host as this device and a key is pressed on the other keyboard while this device is transmitting, then the data transmitted by this device may get corrupted.

Because of potential “data interleave” issues associated with the USB Keyboard interface, MagTek recommends that this product should only be used if the application requires data to be provided via the keyboard input.

Third, MICRSafe can communicate with a device other than the host, for instance a POS terminal, through an auxiliary RS-232 interface. All data is transmitted as ASCII characters (See **Appendix E ASCII Codes**). MICRSafe has the capability of supporting some hardware handshaking signals (see section **4 Legacy Commands**). Depending on POS connection port, a ‘Mini DIN 9 Pin’ or a ‘DB9’ cable can be used to connect. Communication in this mode is one-way only, outputting data from MICRSafe to POS terminal. Baud rate is always set to 9600 bps.

1.1 Features

- This device incorporates a MICR Reader with a 3-Track MagneSafe MSR.
- The three track MSR automatically detects different card formats: ISO (International Standards Organization), CDL (California Drivers License), or AAMVA (American Association of Motor Vehicle Administrators).
- Small footprint.
- Automatic parsing of MICR fields: transit, account, etc.
- Extensive list of formats to transmit MICR data.
- Optional error/status reporting for check reading.
- Optional TDES DUKPT encryption of MICR and Card data.
- Reads E13-B and CMC-7 MICR fonts.
- Automatically goes into sleep mode when not in use; meets EnergyStar requirements.
- EMF noise detection
- Compatible with USB specification Revisions 2.0 and 1.1
- Compatible with HID specification Versions 2.0 and 1.1
- USB communications with the host may occur via a Virtual COM port or by HID/keyboard emulation.
- Communications with third party systems may occur via an auxiliary RS-232 interface or using standard Windows HID drivers. No third party device driver is required.

1.2 Accessories

Accessories available for MICRSafe are as follows:

- Standard USB cable, Part Number 22553301
- Optional auxiliary RS-232 cable, Part Number 22517584 /22517509
- DC Power Adapter with Cable, 120VAC to 12 VDC, 1 Amp, Part Number 64300118 (64300121 for international customers)
- MICR Reader Cleaning Card, Part Number 96700006
- Sample Checks, Part Number 96530005
- MICRbase Setup Program, Part Number 22000021

1.3 Software Drivers Required

If you are using the HID device, the standard HID and Keyboard drivers that come with an operating system are usually all that is needed. For example, the Windows operating system provides all the drivers needed to communicate to the device, unless you requested that the factory configure your MICRSafe device(s) to use a VCOM port. In that case, you would need to download the VCOM driver from the MagTek website. This driver allows a USB device such as MICRSafe to appear as an additional COM port available to the PC, enabling application software to access the USB device as if it were connected via a standard COM (RS-232) port.

1.4 Reference Documents

- Axelson, Jan. ***USB Complete, Everything You Need to Develop Custom USB Peripherals***, 1999. Lakeview Research, 2209 Winnebago St., Madison WI 53704, 396pp., *http://www.lvr.com*
- MICRbase setup program for MICR readers Software and Operation P/N ***D99875102***
- ***USB Human Interface Device (HID) Class Specification*** Version 1.1
- ***Universal Serial Bus (USB): HID Usage Tables*** Version 1.12 (1/21/2005)
- ***USB (Universal Serial Bus) Specification, Version 1.1,*** Copyright ©1998 by Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, NEC Corporation
- ***ANS X9.24-2004 Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques***
- USB Implementers Forum, Inc., *www.usb.org*

1.5 Specifications

Appendix A provides MICRSafe's specifications.

2 Installation

This section provides steps for installing MICRSafe.

2.1 Requirements

The following is required for installation:

- MICRSafe Device, Part Number 22551001 (VCOM) or 22551002 (HID)
- Standard USB cable, Part Number 22553301
- Optional auxiliary RS-232 cable, Part Number 22517584
- DC Power Adapter with Cable, 120VAC to 12 VDC, 1 Amp, Part Number 64300118 (64300121 for international customers)
- MICRbase software, included in the CD Package Part Number 30037855, or can be downloaded from the device's support page on www.magtek.com.
- VCOM Driver for MICRSAFE, included in the CD Part Number 30037903 or can be downloaded from: http://www.magtek.com/support/software/programming_tools/, under 'MICRSafe VCOM'.

2.2 Procedure

Perform the following steps:

1) Connect the interface cable's USB A connector to the PC.
2) Connect the interface cable's USB B connector to MICRSafe.
3) Connect the DC power adapter's jack to the plug on MICRSafe.
4) Connect the DC power adapter's plug to a properly grounded AC socket-outlet.
5) The first time the device is connected to the PC, Windows needs to install the USB driver. See the instructions below.
6) The LED indicator on MICRSafe should turn solid green. The LED indicator is located to the left of the slot where the check is first inserted for reading.

⚠ CAUTION
Do not place MICRSafe within 6 inches of a computer monitor or power supply. These devices may cause undesirable interference with check reading operation.

2.3 USB Driver Installation (Windows)

On hosts with the Windows operating system, the first time MICRSafe is plugged into a specific USB port, Windows opens a dialog box to guide you through the process of installing a driver; follow the instructions. Windows installs the driver that is used for HID keyboard devices, which is a basic component of all modern versions of Windows. Sometimes, Windows finds all the files it needs. Other times, Windows needs to know the location of the files it needs. If Windows prompts for the file locations, insert the CD that was used to install Windows on your PC, and point Windows to the CD's root directory. Windows should find all the files it needs there. After installation has been completed once, the driver installation process will not recur unless the device is subsequently plugged into a different USB port. After installation, the device can communicate with software on the host, such as MICRbase (P/N 22000021) to read checks, credit cards, and setup configurations.

3 Operation

This section contains check and card reading procedures and LED indicator states.

3.1 Check Reading Procedure

1) Orient the check so the MICR line is down and the printed side faces the center of MICRSafe as shown in **Figure 3-1**.

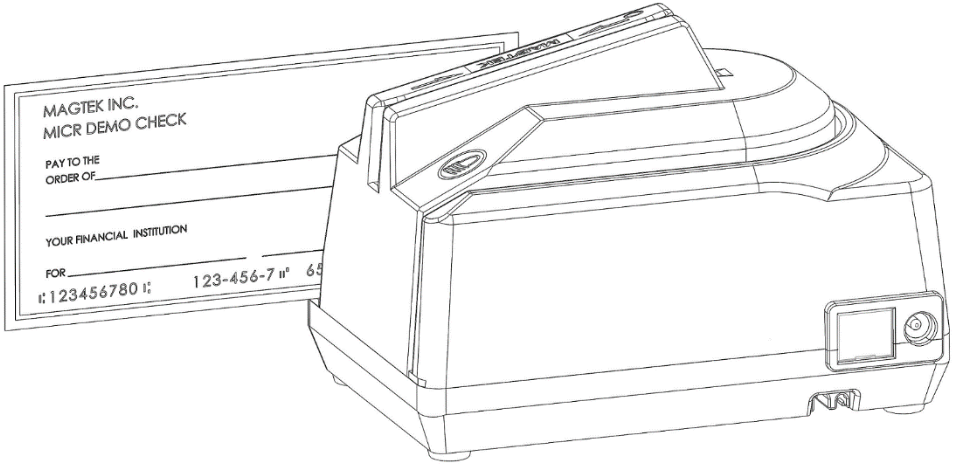


Figure 3-1 - Check Orientation

2) Drop the check so the leading edge is in the open slot.
3) When MICRSafe detects a check is present, its motor turns on. Gently urge the check forward until the device engages with the check, then let go. MICRSafe transports the check around the check path and eject it through the other side.
4) After MICRSafe reads the check, it transmits the data as specified by the parameters described in section **4 Legacy Commands**.

3.2 Card Swipe Procedure

An operator can swipe a magnetic stripe card through the MSR in either direction, but the magnetic stripe must be oriented in only one direction, shown in **Figure 1-1**. MICRSafe transmits raw card data (“as is” on the card) for all tracks that have been enabled using the HW (Hardware) command (section **4 Legacy Commands**).

The MSR is capable of reading ISO, AAMVA, and CDL encoded cards. The MSR automatically detects card formats when the ID Card Decoding option is enabled using **Command ‘HW’ - Hardware**.

3.3 LED Indicators

Table 3-1 describes the LED indicator conditions for check and card reading operations. The LED indicator for check reading is located to the left of the slot where the check is first inserted for reading. The LED indicator for card reading is located on the upper left side of the MSR rail.

Table 3-1 - LED Indicators

LED INDICATOR	DESCRIPTION
OFF	Power off
SOLID GREEN	Ready to read check or card
OFF→ SOLID RED	Check or card read error

LED INDICATOR	DESCRIPTION
OFF → SOLID GREEN	Good read
FLASH RED/GREEN	Data sensor blocked (motor does not run) See **Appendix D Troubleshooting Guide**.
FLASH RED	Motor sensor blocked (motor does not run) See **Appendix D Troubleshooting Guide**.

4 Legacy Commands

This section describes the use of commands and programmable options available across all MICRSafe devices. Some models offer additional commands and settings, which are documented in section **5 USB Communication**.

Note: All options described below can be factory set as specified by the purchaser when ordering.

You may use either of two methods to execute MICRSafe commands: Insta-Change checks or the MICRbase Setup Program for Windows (see *NOTE below).

4.1 Insta-Change Checks

The first way to change device settings is to use an Insta-Change checks, which is more practical for most applications. Insta-Change checks are MICR-encoded documents that contain commands and options that set the parameters of the MICR Reader. A single Insta-Change check may contain multiple commands and options.

To operate Insta-Change checks, install MICRSafe as described in section **2 Installation** and watch the LED indicator, then scan the Insta-Change check through MICRSafe just like a standard check. After MICRSafe successfully reads an Insta-Change check, it blinks the LED indicator green. The device then automatically loads the options from the check. If the LED indicator turns red, the read was not successful. If the LED is blinking red and green, the Insta-Change check is not formatted correctly. Try again or use a different Insta-Change check.

To obtain Insta-Change checks, contact your MagTek representative and specify what options should be encoded onto it.

4.2 MICRbase Setup Program for Windows

The MICRbase Setup Program for Windows (P/N 22000021) allows an advanced operator to control all the programmable options available in MICRSafe.

MICRbase provides a graphical, user-friendly interface that hides the complexities involved in manually entering commands. The operator is not required to know the specific commands or the detailed data associated with each command. However, the program still allows manual entry of commands for advanced operators, and it also displays data from cards and checks that the device reads. If your MICRSafe is using a VCOM port, the legacy MICR commands found in this section may be sent to the device exactly as described in this section. However, if your MICRSafe is using an HID interface, you must send the legacy commands to the device using **Command 0x07 - Send Legacy Command**.

MICRbase version 8.3 or newer works with MICRSafe in either configuration. When MICRSafe is configured as a Keyboard Emulation HID device, you may also program the device using the USB Swipe and Insert Reader demo (version 1.08 or newer). For details about using that tool, see section **5 USB Communication**.

For details and examples of how to use MICRbase, see P/N ***D99875102***. For more detailed information also refer to the Readme.txt file that comes with this program. You can download the MICRbase program from www.magtek.com.

4.3 Command Format

You must use the following format when entering the commands manually:

```
[COMMAND][DATA]<CR>
```

where:

- `[COMMAND]` is a string of alpha characters (usually 2 or 3 bytes in length).
- `[Data]` is optional as described below for each command.
- `<CR>` is always required.
- All characters are ASCII.
- No spaces, brackets, or angle brackets required.

If MICRSafe is using an HID Keyboard Emulation interface, all legacy MICR commands found in this section must be sent to the device using **Command 0x07 - Send Legacy Command**.

4.4 Command ‘SWB’ - Switch B

The SWB command controls the message format, shown in **Table 4-1**. The data for this command consists of 8 ASCII bits (“0” = hex 30 and “1” = hex 31).

To execute, send the SWB command as follows:

`SWB01010101<CR>` (with data)
or
`SWB<CR>` (without data)

When sending data, all 8 bits must be provided. MICRSafe executes the command but it does not reply. The new settings become effective immediately.

To make this setting permanent, the host MUST send **Command ‘SA’ - Save**.

If no data is sent, MICRSafe responds with the current settings for SWB.

Table 4-1 - SWB Command

Bit								Parameters
7	6	5	4	3	2	1	0	
							0	<LF>: No
							1	<LF>: Yes
						0		<CR>: No
						1		<CR>: Yes
					0			<ETX>: No
					1			<ETX>: Yes
				0				<ESC>: No
				1				<ESC>: Yes
			0					<STX>: No
			1					<STX>: Yes
		0						Send Data After Error?: No
		1						Send Data After Error?: Yes
	0							Send Status After Data?: No
	1							Send Status After Data?: Yes
0			0	0	0	0	0	Comm Mode: 0 - Data Only
1			0	0	0	0	0	Comm Mode: 1 - Data<CR>
0			0	0	0	0	1	Comm Mode: 2 - Data<LF>
0			0	0	0	1	1	Comm Mode: 3 - Data<CR><LF>
0			0	1	0	0	0	Comm Mode: 4 - <ESC>Data

Bit								Parameters
0			0	1	0	1	0	Comm Mode: 5 - <ESC>Data<CR>
0			1	0	1	0	0	Comm Mode: 6 - <STX>Data<ETX>
1			0	0	0	0	1	Comm Mode: 7 - <STX>Data<ETX><LRC>
0			1	1	1	1	1	Comm Mode- 8 – All controls
0			0	0	1	0	1	Comm Mode- 9- Data<ETX><LF>
0			1	0	0	1	0	Comm Mode- 10- <STX>Data<CR>

4.4.1 Control Characters and MICR Data

Control Characters may be added to MICR Data messages. MICRSafe inserts any control characters selected using this command into outgoing formatted MICR Data messages in the following sequence:

```
<STX> <ESC> data <ETX> <CR> <LF>
```

The control characters, descriptions, and hex values are shown in **Table 4-2**.

Table 4-2 - Control Characters

Control Character	Description	Hex Value
<STX>	Start of Text	02
<ESC>	Escape	1B
<ETX>	End of Text	03
<CR>	Carriage Return	0D
<LF>	Line Feed	0A

For example, if the <STX> and <CR> options are set to YES, a MICR Data message from MICRSafe looks like this:

```
MICR Data: <STX>data<CR>
```

4.4.2 Control Characters and Card Data

If the card reader's head is set to security level 2, then the same control characters may also be added to Card Data messages, but they are applied to each track individually. For example, if the <STX> and <ETX> options are set to YES, a Card Data message from MICRSafe looks like this:

```
Card Data: <STX>[TK1 data]<ETX><STX>[TK2 data]<ETX><STX>[TK3
data]<ETX>
```

Please note again that card reader heads that are set to security level 1 and 3 do not support this control characters.

4.4.3 Communication Modes

The selection of Comm modes is a quick way of selecting multiple Control Characters. For instance, to send a carriage return/line feed pair after the data, you can specify Comm Mode 3.

Comm Mode 7, also known as Packet Mode, calculates an LRC (Longitudinal Redundancy Check), and appends it to the data message. Also, if a <NAK> (hex 15) character is received in this mode, MICRSafe resends the last message.

4.4.4 Send Data After Error

The Send Data After Error option specifies whether MICRSafe returns data to the host after a read error. If YES is selected and MICRSafe detects a read error, MICRSafe still sends the data back to the host. If NO is selected and MICRSafe finds an error, it discards the data and sends nothing. The error conditions are listed in **Table 4-3**.

4.4.5 Send Status After Data

The Send Status After Data option causes MICRSafe to append a two-digit error/status code to the end of the MICR data. For most formats (See **Appendix B**), the error/status code is always preceded by a forward slash (/). The error/status codes are listed in **Table 4-3**.

For example, if a Canadian check (code 08) is read and has no errors, and the MICR data is "1234567890", then the message from MICRSafe looks like this:

```
MICR Data:  1234567890/08
```

The status code is always at the end of the data, not the end of the message. For example, using the above conditions, with the message format set to send <STX> and <ETX>, the message from MICRSafe looks like this:

```
MICR Data:  <STX>1234567890/08<ETX>
```

Table 4-3 - Error and Status Codes

Priority	Code	Type	Description
9	01	Error	No MICR data: no transit and no account found
8	09	Status	Mexican check
7	08	Status	Canadian check
6	05	Error	Transit error: No transit, bad character, bad length, bad check digit
5	07	Error	Account error: No account, bad character
4	04	Error	Check # error: Bad character in check number
4	04	Status	No check number
3	03	Status	Low MICR signal, good read
2	10	Status	Business check
1	11	Status	Amount field present
0	00	Status	Good read

The LED indicator turns red on all error conditions. The absence of a check number is not considered an error. If a multiple error condition occurs, the device reports the error or status code with the highest priority. All unreadable MICR characters are transmitted as an "?" ASCII character (hex 3F), except for Format 00xx (See **Appendix B**).

4.5 Command ‘SWC’ - Switch C

The SWC command controls miscellaneous functions, shown in **Table 4-4**. The data for this command consists of 8 ASCII bits (“0” = hex 30 and “1” = hex 31).

To execute, send the SWC command as follows:

`SWC01010101<CR>` (with data)
or
`SWC<CR>` (without data)

When sending data, all 8 bits must be provided. MICRSafe executes the command but does not reply. The new settings become effective immediately.

To make this setting permanent, the host MUST send **Command ‘SA’ - Save**.

If no data is sent, MICRSafe responds with the current settings for SWC.

Table 4-4 - SWC Command

Bits								Parameters
7	6	5	4	3	2	1	0	
							0	CMC-7 Character Set: No
							1	CMC-7 Character Set: Yes
					0	0		Invalid Commands: ?<CR>
					0	1		Invalid Commands : No Reply (Header Required)[2]
					1	0		Invalid Commands: No Reply (No Header Required)
					1	1		Ignore all Commands
				0				Reserved
			0					Data Header: No[1]
			1					Data Header: Yes[1]
		0						Card Data Message: Multiple[1]
		1						Card Data Message: Single[1]
0	0							These bits are always set to 0 but must be included.

1) For card data, this parameter does not apply when the card reader head is set to security level 1 or 3
2) ‘Header Required’ means all commands must be preceded by a GS character (0x1D).

4.5.1 CMC-7 Character Set

If NO is selected, MICRSafe only reads E13-B characters. When YES is selected, MICRSafe reads both CMC-7 and E13-B characters (see **Appendix C Check Reading**). However, MICRSafe only outputs raw data ("as is" on the check) for checks with CMC-7 characters.

4.5.2 Invalid Command Response

Invalid command response is the action MICRSafe takes upon receipt of a command it does not recognize. It can also be used to stop MICRSafe from receiving any more commands.

The first option "**?<CR>**" is the default. If MICRSafe receives an unrecognized command, it returns a question mark and carriage return to the host. MICRSafe then returns to an idle state and waits for further commands or check/credit card reads.

For the second option, "no reply - header required," MICRSafe only executes commands preceded by a GS ASCII character (hex 1D). It ignores all other commands. Also, MICRSafe does not reply to invalid commands.

For the third option, "no reply – no header required," MICRSafe executes all valid commands, but does not reply to invalid commands.

The fourth option, "ignore all commands," causes MICRSafe to ignore any further commands. Even the SA (Save) command is ignored and therefore this fourth option is only temporary. To make this option permanent or to reset it, you must use an Insta-Change check.

4.5.3 Data Header

If YES is selected, a single character header precedes the data. For MICR data, the message is transmitted as follows:

MICR data: 'C'[data]

For card data, the header position on the message is controlled by the **Card Data Message** parameter. The message may be transmitted as follows:

- Multiple Message: **'M'[TK1]'M'[TK2]'M'[TK3]**
- Single Message: **'M'[TK1] [TK2] [TK3]**

It is important to note that the Data Header precedes the data and not the message. For example, if <STX>, <ETX> and Data Header are set to YES, a MICR data message is transmitted as follows:

MICR data: <STX>'C'[data]<ETX>

4.5.4 Card Data Message

This option determines the structure of the output message for the individual tracks when a credit card is read. If Multiple is selected, the Control Characters (see SWB command, above) and Data Header (see Data Header section, above) are added to each track individually. On the other hand, if Single is selected, all available tracks are lumped together into a single message. For example, if <STX>, <ETX> and Data Header are set to YES, the output message may be transmitted as follows (note, this option applies only when the card reader head is set to security level 2):

- Multiple Message: **<STX>'M'[TK1]<ETX><STX>'M'[TK2]<ETX><STX>'M'[TK3]<ETX>**
- Single Message: **<STX>'M'[TK1] [TK2] [TK3]<ETX>**

4.6 Command ‘HW’ - Hardware

This command controls miscellaneous hardware options, as shown in **Table 4-5**. The data for this command consists of 8 ASCII bits (“0” = hex 30 and “1” = hex 31).

To execute, send the HW command as follows:

HW 01010101<CR> (with data)
or
HW <CR> (without data)

When sending data, all 8 bits must be provided. MICRSafe executes the command but does not reply. The new settings become effective immediately.

To make this setting permanent, the host MUST send **Command ‘SA’ - Save**.

If no data is sent, MICRSafe responds with the current settings for HW.

Table 4-5 - HW Command

7	6	5	4	3	2	1	0	PARAMETERS
					0			Track 3: Disable[1]
					1			Track 3: Enable[1]
				0				Track 2: Disabl[1]*
				1				Track 2: Enable[1]
			0					Track 1: Disable[1]
			1					Track 1: Enable[1]
		0						ID Card decoding: Disable[1]
		1						ID Card decoding: Enable[1]
						0		EMF detect: Yes
						1		EMF detect: No
0	0						0	These bits are always set to 0
1) This option applies only when the card reader head is set to security level 2								

4.6.1 Disable / Enable Tracks

Each track can be enabled or disabled individually. The tracks are always transmitted in ascending order: TK1, TK2, TK3. For example, if TK1 and TK3 are enabled and TK2 is disabled, the MSR transmits TK1, TK3.

4.6.2 ID Card Decoding

The MSR has two modes of operation. In the first mode, ID Card decoding disabled, the MSR only reads ISO encoded cards. In the second mode, ID Card decoding enabled, the MSR reads and automatically detects ISO, AAMVA, and CDL encoded cards. When a card is swiped, the LED indicator turns red and indicates an error if any of the enabled tracks read is incompatible with the selected mode of operation. TK2 is a standard track for all types of cards.

4.6.3 EMF Detect

The EMF Detect option allows the MICR Reader, when idle, to monitor EMF interference in its immediate environment. If YES is selected, the LED indicator blinks red/green when MICRSafe detects a signal with amplitude large enough to affect check reading. If NO is selected, MICRSafe does not monitor nor indicate the presence of EMF interference.

4.7 Command 'FC' - Format Change

Formats are used by MICRSafe to process and transmit the MICR fields. The format command allows the selection of a format from the Format List (see **Appendix B Format List**). The data for this command consists of 4 digits (ASCII characters 0-9). To execute, send the command as follows:

```
FCXXXX<CR>  (with data)
or
FC<CR>      (without data)
```

When sending data, all 4 digits must be provided. MICRSafe executes the command but does not reply. The new settings become effective immediately.

To make this setting permanent, the host MUST send **Command 'SA' - Save**.

If no data is sent, MICRSafe responds with the current format number.

4.8 Command 'VR' - Version

The Version command gives the current software revision in the MICR Reader. To execute, send the VR command followed by a carriage return as follows:

```
VR<CR>
```

MICRSafe responds as follows:

```
MICR data:  [software revision]<CR>
```

4.9 Command 'SA' - Save

All changes are considered temporary until the host sends the Save command, which signals the device to write all changes to permanent memory. MICRSafe executes this command but does not send a response. To execute, send the SA command followed by a carriage return as follows:

```
SA<CR>
```

4.10 Command 'RS' - Reset

The Reset command resets MICRSafe firmware to the normal operating state of waiting for a check or card to be read. The command also resets the serial port to the most recently saved settings. To execute, send the RS command followed by a carriage return as follows:

```
RS<CR>
```

4.11 Command 'DM' - Disable MICR

This command disables the document reading function and turns off the LED. Communications are not affected. The motor does not turn on when a document is inserted. To execute, send the DM command followed by a carriage return as follows:

```
DM<CR>
```

4.12 Command 'EM' - Enable MICR

This command enables the document reading function, and the LED turns green. To execute, send the EM command followed by a carriage return as follows:

EM<CR>

4.13 Command 'KS' - Enable Keystroke

This command determines if MICRSafe sends responses to the host via the USB interrupt pipe or as Keystrokes (that appear to the host application as if typed on a keyboard). To execute, send the KS command followed by a carriage return as follows:

KS00<CR> (to send responses to the host via the USB interrupt pipe)
or
KS01<CR> (to send responses as keystrokes)

4.14 Command 'SLP' - Sleep Mode

This command puts MICRSafe into sleep mode for the specified time interval. While in sleep mode, the green LED dims to half brightness. To execute, send the SLP command followed by a carriage return as follows:

SLPB4<CR> (to put MICRSafe in sleep mode after thirty minutes of idle state)

NOTE: The two characters following "SLP" contain the Hex value of the time before the device goes into sleep mode, with time measured in ten-second intervals. Hex FF (numeric 255) is the maximum time you can specify; 255 ten second intervals equal 42 minutes.

4.15 Command 'RD' - Enable Auxiliary Port

This command determines if MICRSafe sends responses to the host via the auxiliary RS-232 port. To execute, send one of the following RD commands followed by a carriage return:

RD00<CR> (to disable the use of the auxiliary RS-232 port)
or
RD01<CR> (to enable the use of the auxiliary RS-232 port for transmitting MSR data only)
or
RD02<CR> (to enable the use of the auxiliary RS-232 port for transmitting check data only)
or
RD03<CR> (to enable the use of the auxiliary RS-232 port for transmitting both MSR and check data)

The new settings become effective immediately.

To make this setting permanent, the host MUST send **Command 'SA' - Save**.

If no data is sent, MICRSafe responds with the current value of RD.

4.16 Command 'CHKCNT' - Check Count

This command returns the number of checks that have been processed by the device. To execute, send the CHKCNT command followed by a carriage return as follows:

CHKCNT<CR>

5 USB Communication

This device conforms to the USB specification revisions 2.0 and 1.1. This device also conforms to the Human Interface Device (HID) class specification versions 2.0 and 1.1. The device communicates to the host as a HID keyboard device. The latest versions of the Windows operating systems come with a standard Windows USB HID keyboard driver. This section pertains only to the HID Keyboard Emulation configuration. For character mapping and character conversion, you may reference **Appendix E, Appendix F**, and **Appendix G** for ASCII codes and USB Usage ID definitions.

This is a full speed USB device. This device has a number of programmable configuration properties. These properties are stored in non-volatile memory. These properties can be configured at the factory or by the end user. The device has an adjustable endpoint descriptor polling interval value that can be set to any value in the range of 1ms to 255ms. This property can be used to speed up or slow down the keyboard data transfer rate. The device also has an adjustable serial number descriptor. More details about these properties can be found later in this document in the command section.

The device goes into suspend mode when directed to do so by the host. The device wakes up from suspend mode when directed to do so by the host. The device does not support remote wakeup.

This device is powered from the USB bus. The vendor ID is **0x0801** and the product ID is **0x2251**.

5.1 USB Device Commands and Responses

5.1.1 About Host Software

If MICRSafe is configured as an HID device, it can be used with existing host software that acquires card data via keyboard input. It is easy to develop new host software that communicates with the device, and such software can be developed using compilers such as Microsoft's Visual Basic or Visual C++. To demonstrate this device's card reading capabilities, any software that accepts keyboard input (for instance Windows Notepad) can be used.

5.1.2 About Card Data and MICR Data

MICRSafe converts magnetic stripe card data and MICR data to ASCII and transmits it to the host as if it had been typed on a keyboard.

Because of potential "data interleave" issues associated with the USB Keyboard interface, MagTek recommends that you do not depress the keyboard while swiping a card or scanning a check. If previous applications were based upon RS-232 serial interface on a Windows operating system, it is recommended that you use MagTek's MINI MICR USB Virtual COM Port product.

The device's programmable configuration options affect the format of the card and MICR data. Refer to the legacy commands section for a description of how the card and MICR data is formatted. Some of the properties in this section also affect the format of the card and MICR data.

The device sends all data in uppercase, regardless of the state of the caps lock key on the keyboard.

⚠ CAUTION
If another keyboard is connected to the same host as this device and a key is pressed on the other keyboard while this device is transmitting, then the data transmitted by this device may get corrupted.

5.1.3 About Low level Communication

During normal operation, the device acts like a keyboard, and the host operating system takes care of low level communication with the device so the host software developer is not burdened with these details. The next few sections provide details about how to communicate with the device to send commands and receive responses, and to change programmable configuration properties. These details are included as a reference only. Most users do not need to know these details because the device will be configured at the factory or by a program supplied by MagTek. Most users may wish to skip the next few sections and resume reading the details about the device's configuration properties.

MagTek strongly recommends application software developers become familiar with the HID specification and the USB specification before attempting to communicate directly with this device. This document assumes that the reader is familiar with these specifications. These specifications can be downloaded free from www.usb.org.

5.1.4 About HID Usages

HID devices send data in reports. Elements of data in a report are identified by unique identifiers called usages. The structure of the device's reports and the device's capabilities are reported to the host in a report descriptor. The host usually gets the report descriptor only once, right after the device is plugged in. The report descriptor usages identify the device's capabilities and report structures. For example, a device could be identified as a keyboard by analyzing the device's report descriptor. Usages are four byte integers. The most significant two bytes are called the usage page and the least significant two bytes are called usage IDs. Usages that are related can share a common usage page. Usages can be standardized or they can be vendor-defined. Standardized usages, such as usages for mice and keyboards, can be found in the HID Usage Tables document and can be downloaded free at www.usb.org. Vendor-defined usages must have a usage page in the range 0xFF00 – 0xFFFF. All usages for this device use the standard HID keyboard usages or vendor-defined magnetic stripe reader usage page 0xFF00. The vendor-defined usage IDs for this device are defined in the following table. The usage types are also listed. These usage types are defined in the ***HID Usage Tables*** document.

Magnetic Stripe Reader usage page 0xFF00

Usage ID (Hex)	Usage Name	Usage Type	Report Type
20	Command message	Data	Feature

5.1.5 About the Report Descriptor

The HID report descriptor is structured as follows:

Item	Value(Hex)
Usage Page (Generic Desktop)	05 01
Usage (Keyboard)	09 06
Collection (Application)	A1 01
Usage Page (Key Codes)	05 07
Usage Minimum (224)	19 E0
Usage Maximum (231)	29 E7
Logical Minimum (0)	15 00
Logical Maximum (1)	25 01
Report Size (1)	75 01
Report Count (8)	95 08
Input (Data, Variable, Absolute)	81 02
Report Count (1)	95 01
Report Size (8)	75 08
Input (Constant)	81 03
Report Count (5)	95 05
Report Size (1)	75 01

Item	Value(Hex)
Usage Page (LEDs)	05 08
Usage Minimum (1)	19 01
Usage Maximum (5)	29 05
Output (Data, Variable, Absolute)	91 02
Report Count (1)	95 01
Report Size (3)	75 03
Output (Constant)	91 03
Report Count (6)	95 06
Report Size (8)	75 08
Logical Minimum (0)	15 00
Logical Maximum (101)	25 66
Usage Page (Key Codes)	05 07
Usage Minimum (0)	19 00
Usage Maximum (101)	29 66
Input (Data, Array)	81 00
Logical Maximum (255)	26 FF 00
Usage Page (vendor-defined (MSR))	06 00 FF
Usage (command data)	09 20
Report Count	95 18
Feature (Data, Variable, Absolute, Buffered Bytes)	B2 02 01
End Collection	C0

5.1.6 About Command Processing

Firmware in the device's magnetic stripe reader normally handles card swipes, gets MagnePrint data, and encrypts the data. It then passes the output to firmware in MICRSafe for transmission to the host. The firmware controls all functions relating to check reading and handles all communication between the host and the device. Thus, all commands are received by the device's firmware, which, unless instructed otherwise, attempts to process the command. For this reason, commands that affect card reading only (which are unknown to the firmware) must be distinguished from commands which the firmware is equipped to handle. The convention adopted to enable this necessary distinction is to prefix the data portion of a card reader command with the string literal `MSR`. This signals the firmware to *not* process the command, and instead pass the command to the MSR firmware.

5.1.7 About Command and Response Formats

Command requests and responses are sent to and received from the device using feature reports. Command requests are sent to the device using the HID class specific request **Set Report**. The response to a command is retrieved from the device using the HID class specific request **Get Report**. The requests are sent over the default control pipe. When a command request is sent, the device NAKs the Status stage of the **Set Report** request until the command is completed. This insures that as soon as the **Set Report** request is completed, the **Get Report** request can be sent to get the command response. The usage ID for the command message was shown previously in the Usage Table.

The following table shows how the feature report is structured for command requests:

Offset	Field Name
0	**Command Number**
1	**Data Length**
2..23	**Data**

The following table shows how the feature report is structured for command responses.

Offset	Field Name
0	**Result Code**
1	**Data Length**
2..23	**Data**

5.1.7.1 Command Number

This one-byte field contains the value of the requested command number, which is included in the section heading and description for each command in this section.

5.1.7.2 Data Length

This one-byte field contains the length of the valid data contained in the Data field.

5.1.7.3 Data

This multi-byte field contains command data, if any. Note that the length of this field is fixed at 22 bytes. Valid data should be placed in the field starting at offset 2. Because the HID specification requires reports be a constant length, the host must pad any remaining space after command data with zeroes, including situations where there is no data to send at all.

5.1.7.4 Result Code

This one-byte field contains the value of the result code. There are two types of result codes: generic result codes and command-specific result codes. Generic result codes always have the most significant bit set to zero. Generic result codes have the same meaning for all commands and can be used by any command. Command-specific result codes always have the most significant bit set to one. Command-specific result codes are defined by the command that uses them. The same code can have different meanings for different commands. Command-specific result codes are defined in the documentation for the command that uses them. Generic result codes are defined in the following table.

Value	Result Code	Description
0	SUCCESS	The command completed successfully.
1	FAILURE	The command failed.
2	BAD_PARAMETER	The command failed due to a bad parameter or command syntax error.

5.1.8 Command 0x00 and Command 0x01 - Get Property and Set Property

In addition to settings that implicit in the device's **Legacy Commands**, the device's behavior can be further customized using explicit **Properties**, which this manual describes starting with section **5.2.1 Property 0x00 - Software ID**.

The host can use the **Get Property** command, Command ID **0x00**, to read the current value of any configuration property, and can use the **Set Property** command, Command ID **0x01**, to set a new value for any configuration property.

NOTICE

Before using either the Set Property or Get Property command, the host or operator must send *Command 'SA' - Save* **if they have changed any legacy configuration settings (for example, using** *Command 'FC' - Format Change***), which directs the device to permanently store the legacy configuration settings.**

If the host software or operator sends Set Property or Get Property without saving legacy settings first, the device will not preserve those settings, and will immediately overwrite them with the currently saved legacy settings.

The Get and Set Property command data fields for the requests and responses are structured as follows:

Get Property Request Data

Data Offset	Value
0	Property ID

Get Property Response Data

Data Offset	Value
0 – n	Property Value

Set Property Request Data:

Data Offset	Value
0	Property ID
1 – n	Property Value

Set Property Response Data: None

The result codes for the Get and Set Property commands can be any of the codes listed in the generic result code table.

Property ID is a one-byte field that contains a value that identifies the property. The list of all properties and their corresponding Property IDs is provided in this manual beginning with section **5.2.1 Property 0x00 - Software ID**.

Property Value is a multiple-byte field that contains the value of the property. The number of bytes in this field depends on the type of property and the length of the property. The following table lists all of the property types and describes them. The list of all properties is provided beginning with section **5.2.1 Property 0x00 - Software ID**.

Property Type	Description
Byte	This is a one-byte value. The valid values depend on the property.
String	This is a multiple byte ASCII string. Its length can range from zero to a maximum length that depends on the property. The value and length of the string does not include a terminating NUL character.

5.1.9 Command 0x02 - Reset Device

Command number:	2
Data structure:	No data is sent with this command
Result codes:	0 (success)

The host uses this command to reset the device’s USB CPU. This command can be used to make previously changed properties take effect without having to unplug and then plug in the device. When the device resets it automatically does a USB detach followed by an attach. After the host sends this command, it should close the USB port, wait a few seconds for the operating system to handle the device detach followed by the attach, then re-open the USB port before trying to communicate further.

Example Request (Hex):

Cmd Num	Data Len	Data
02	00	

Example Response (Hex):

Result Code	Data Len	Data
00	00	

5.1.10 Command 0x03 - Get Key Map Item

Command number: 3

The host uses this command to get a key map item from the active key map currently selected by **Property 0x11 - Active Key Map**. Data from a card or a check is a sequence of ASCII characters (see **Appendix E**), and the key map specifies which keystrokes the device sends to the host to represent each character. Each of the specified keystrokes is a combination of a **USB key usage ID** and a **USB key modifier byte**: The key usage ID is a unique value assigned to every keyboard key (see **Appendix E**), and the key modifier byte indicates whether any combination of the right or left **Ctrl**, **Shift**, **Alt** or GUI keys are pressed at the same time (see **Appendix G**).

When a key map entry specifies key usage ID and the key modifier byte both set to 0xFF, the device sends the ALT ASCII code instead of the key map values. The ALT ASCII code is a keypress combination consisting of the decimal value of the ASCII character combined with the ALT key modifier. For example, to transmit the ASCII character '?' (063 decimal), the device sends keypad '0' combined with left ALT key modifier, then keypad '6' combined with the left ALT key modifier, then keypad '3' combined with the left ALT key modifier.

Request Data

Offset	Field Name	Description
0	ASCII value	Value of the ASCII character to be retrieved from the key map. This can be any value between 0 and 127 (0x7F). For example, to retrieve the key map item for ASCII character '?' (card data end sentinel) use the ASCII value of '?' which is 63 (0x3F).

Response Data

Offset	Field Name	Description
0	Key Usage ID	The value of the USB key usage ID that is mapped to the given ASCII value. For example, for the United States keyboard map, usage ID 56 (0x38) (keyboard / and ?) is mapped to ASCII character '?'.
1	Key Modifier Byte	The value of the USB key modifier byte that is mapped to the given ASCII value. For example, for the United States keyboard map, modifier byte 0x02 (left shift key) is mapped to ASCII character '?'.

Example Request (Hex)

Cmd Num	Data Len	Data
03	01	3F

Example Response (Hex)

Result Code	Data Len	Data
00	02	38 02

5.1.11 Command 0x04 - Set Key Map Item

Command number: 4

The host uses this command to set a key map item in the device’s active key map. For information about the key map, see **Command 0x03 - Get Key Map Item**.

After the host modifies a key map item, the changes take effect immediately. However, to make the changes permanent, the host must send **Command 0x05 - Save Custom Key Map**. To use the new custom key map after a reset or power cycle, the host must set **Property 0x11 - Active Key Map** to custom.

Request Data

Offset	Field Name	Description
0	ASCII value	Value of the ASCII character to be set in the key map. This can be any value between 0 and 127 (0x7F). For example, to set the key map item for ASCII character ‘?’ (card data end sentinel) use the ASCII value of ‘?’ which is 63 (0x3F).
1	Key Usage ID	The value of the USB key usage ID that is to be mapped to the given ASCII value. For example, for the United States keyboard map, usage ID 56 (0x38) (keyboard / and ?) is mapped to ASCII character ‘?’. To change this to the ASCII character ‘>‘ use usage ID 55 (0x37) (keyboard . and >).
2	Key Modifier Byte	The value of the USB key modifier byte that is to be mapped to the given ASCII value. For example, for the United States keyboard map, modifier byte 0x02 (left shift key) is mapped to ASCII character ‘?’. To change this to the ASCII character ‘>‘ use modifier byte 0x02 (left shift key).

Response Data: None

Result codes: 0 (success)

The following example maps the card ASCII data end sentinel character ‘?’ to the ‘>’ keyboard key.

Example Request (Hex):

Cmd Num	Data Len	Data
04	03	3F 37 02

Example Response (Hex):

Result Code	Data Len	Data
00	00	

5.1.12 Command 0x05 - Save Custom Key Map

Command number: 5

This command is used to save the active key map as the custom key map in non volatile memory. The active key map is determined by the active key map property. Once a key map item is modified, the changes take effect immediately. However, the changes will be lost if the device is reset or power cycled. To make the changes permanent, the save custom key map command must be issued. To use the new custom key map after a reset or power cycle, the active key map property must be set to custom.

Request Data: None

Response Data: None

Result codes: 0 (success)

Example Request (Hex)

Cmd Num	Data Len	Data
05	00	

Example Response (Hex)

Result Code	Data Len	Data
00	00	

5.1.13 Command 0x07 - Send Legacy Command

Command number: 7

This command is used to send legacy commands to the device through the USB connection. The device sends up to two responses to this command: The first is the standard USB response, which indicates the device received the USB command successfully. If the legacy command is expected to send a response, it sends it to the host as keystrokes. See section **4 Legacy Commands** for all of the legacy commands the host can send using this wrapper command.

Request Data: Legacy command to be sent including carriage return if any. The legacy command must be converted from ASCII to binary data prior to sending it. For example, the command SWB<CR> must be sent as 53 57 42 0D (hex). The legacy command can be sent one character at a time or all at once up to the maximum size allowed by a set feature report (22 bytes). For example, the command SWB<CR> can also be sent by issuing the **Send Legacy Command** 4 times, once for each of the four characters. In this case, the device does not send a response to the legacy command until after the <CR> is sent.

Response Data: None

Result codes: 0 (success)

Example Request (Hex):

Cmd Num	Data Len	Data (SWB<CR>)
07	04	53 57 42 0D

Example Response (Hex):

Result Code	Data Len	Data
00	00	

Example Legacy Command Response (sent as keystrokes):
SWB=00000000<CR>

5.2 USB Device Properties

This device has a number of programmable configuration **properties**. These properties are stored in non-volatile memory, and can be configured at the factory or by an advanced operator using a program supplied by MagTek. Programming these parameters requires low level communication with the device.

5.2.1 Property 0x00 - Software ID

Property ID: 0
Property Type: String
Length: Fixed at 11 bytes
Get Property: Yes
Set Property: No
Description: This is an 11 byte read only property that identifies the software part number and version for the device's USB CPU. The first 8 bytes represent the part number and the last 3 bytes represent the version. For example this string might be "22827021A01". Examples follow.

Example Get Software ID property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	00

Example Get Software ID property Response (Hex):

Result Code	Data Len	Prp Value
00	01	32 32 38 32 37 30 32 31 41 30 31

5.2.2 Property 0x01 - Serial Num

Property ID: 1
Property Type: String
Length: 0 – 15 bytes
Get Property: Yes
Set Property: Yes
Default Value: The default value is no string with a length of zero.

The value is an ASCII string that represents the device's serial number. This string can be 0 – 15 bytes long. The value of this property, if any, is sent to the host when the host requests the USB string descriptor.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Serial Num property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	04	01	32 32 38 32 37 30 32 31 41 30 31

Example Set Serial Num property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Serial Num property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	01

Example Get Serial Num property Response (Hex):

Result Code	Data Len	Prp Value
00	03	32 32 38 32 37 30 32 31 41 30 31

5.2.3 Property 0x02 - Polling Interval

Property ID: 2
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 1
Description: The value is a byte that represents the device's polling interval for the Interrupt In Endpoint. The value can be set in the range of 1 – 255 and has units of milliseconds. The polling interval tells the host how often to poll the device for keystroke data packets. For example, if the polling interval is set to 10, the host will poll the device for keystroke data packets every 10ms. This property can be used to speed up or slow down the time it takes to send keystroke data to the host. The trade-off is that speeding up the card data transfer rate increases the USB bus bandwidth used by the device, and slowing down the card data transfer rate decreases the USB bus bandwidth used by the device. The value of this property will be sent to the host when the host requests the device's USB endpoint descriptor.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the unit must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Polling Interval property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	02	0A

Example Set Polling Interval property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Polling Interval property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	02

Example Get Polling Interval property Response (Hex):

Result Code	Data Len	Prp Value
00	01	0A

5.2.4 Property 0x04 - Track Data Send Flags

Property ID: 4
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0x00
Description: This property is defined as follows:

ICL	0	0	0	0	0	0	0

ICL 0 – Changing the state of the caps lock key will not affect the case of the data
1 – Changing the state of the caps lock key will affect the case of the data

This property is stored in non-volatile memory, so it will persist when the unit is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

5.2.5 Property 0x0F - ASCII to Keypress Conversion Type

Property ID: 15 (0x0F)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0 (key map)
Description:
The value is a byte that represents the device's ASCII to keypress conversion type. The value can be set to 0 for key map (The active key map is set with the **Active Key Map** property) or to 1 for ALT ASCII code (international keyboard emulation). When the value is set to 0 (key map), data will be transmitted to the host according to the active key map which defaults to the United States keyboard key map. For example, to transmit the ASCII character '?' (063 decimal), the character is looked up in a key map. For a United States keyboard key map, the '/' (forward slash) key combined with the left shift key modifier are stored in the key map to represent the key press combination that is used to represent the ASCII character '?' (063 decimal). When the value is set to 1 (ALT ASCII code), instead of using the key map, an international keyboard key press combination consisting of the decimal value of the ASCII character combined with the ALT key modifier is used. For example, to transmit the ASCII character '?' (063 decimal), keypad '0' is sent combined with left ALT key modifier, next keypad '6' is sent combined with the left ALT key modifier, last keypad '3' is sent combined with the left ALT key modifier. In general, if this device only needs to emulate United States keyboards then this property should be set to 0 (key map).

If the device needs to emulate all countries' keyboards, set this property to 1 (ALT ASCII code). The tradeoff is that the ALT ASCII code mode is slightly slower than key map mode because more key presses need to be transmitted. Some applications are not compatible with ALT ASCII code mode.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set ASCII To Keypress Conversion Type property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	0F	00

Example Set ASCII To Keypress Conversion Type property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get ASCII To Keypress Conversion Type property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	0F

Example Get ASCII To Keypress Conversion Type property Response (Hex):

Result Code	Data Len	Prp Value
00	01	00

5.2.6 Property 0x11 - Active Key Map

Property ID: 17 (0x11)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0 (United States)

The value is a byte that represents the device's active key map, where **0x00 = United States key map** and **0x01 = Custom key map**. The device uses the active key map to convert ASCII data into key strokes (see **Command 0x03 - Get Key Map Item** for details).

Use the United States key map with all hosts that are configured to use United States keyboards. Use the custom key map to communicate with hosts that are configured to use other countries' keyboards. The default custom key map is the same as the United States key map. The host can modify the key map to another country's key map by using **Command 0x03 - Get Key Map Item**, **Command 0x04 - Set Key Map Item**, and **Command 0x05 - Save Custom Key Map**.

To set the device to use a custom key map:

1) Select the appropriate key map to be modified using this property.
2) Reset the device to make this change take effect.
3) Use the commands above to modify and save the active key map.
4) Set this property to custom to use the custom key map.
5) Reset the device to make these changes take effect.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or power cycled for these changes to take effect.

Example Set Active Key Map property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	11	00

Example Set Active Key Map property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Active Key Map property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	11

Example Get Active Key Map property Response (Hex):

Result Code	Data Len	Prp Value
00	01	00

5.2.7 Property 0x1A - Convert from Char A

Property ID: 26 (0x1A)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 255 (0xFF) (None)

The value is a byte that represents the ASCII value of a character transmitted by the device as keystroke data that is to be changed into a string of ASCII values prior to being transmitted by the device as keystroke data. The string of ASCII values that this value will be changed into is contained in the **Convert To String A** property. If the value of this property is set to 0xFF, no characters will be changed into the string. For example, if you would like a carriage return to be sent as an end of text character you could set the **Convert From Char A** property to 0x0D (carriage return) and set the **Convert To String A** property to 0x03 (end of text). If you would like a carriage return to be sent as two carriage returns you could set the **Convert From Char A** property to 0x0D (carriage return) and set the **Convert To String A** property to 0x0D 0x0D (carriage return, carriage return). If you would like a carriage return not to be sent you could set the **Convert From Char A** property to 0x0D (carriage return) and set the **Convert To String A** property to no string.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Convert From Char A property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	1A	FF

Example Set Convert From Char A property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Convert From Char A property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	1A

Example Convert From Char A property Response (Hex):

Result Code	Data Len	Prp Value
00	01	FF

5.2.8 Property 0x1B - Convert to String A

Property ID: 27 (0x1B)
Property Type: String
Length: 0 – 7 bytes
Get Property: Yes
Set Property: Yes
Default Value: The default value is no string with a length of zero.

The value is an ASCII string that represents the device's **Convert To String A** property. This string can be 0 – 7 bytes long. This string is sent in place of the character specified in the **Convert From Char A** property. See the **Convert From Char A** property for more information and examples.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Convert To String A property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	03	1B	0D 0D

Example Set Convert To String A property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Convert To String A property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	1B

Example Get Convert To String A property Response (Hex):

Result Code	Data Len	Prp Value
00	02	0D 0D

5.2.9 Property 0x1C - Convert from Char B

Property ID: 28 (0x1C)
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 255 (0xFF) (None)

The value is a byte that represents the ASCII value of a character transmitted by the device as keystroke data that is to be changed into a string of ASCII values prior to being transmitted by the device as keystroke data. The string of ASCII values that this value will be changed into is contained in the **Convert To String B** property. If the value of this property is set to 0xFF, no characters will be changed into the string. For example, if you would like a carriage return to be sent as an end of text character you could set the **Convert From Char B** property to 0x0D (carriage return) and set the **Convert To String B** property to 0x03 (end of text). If you would like a carriage return to be sent as two carriage returns you could set the **Convert From Char B** property to 0x0D (carriage return) and set the **Convert To String B** property to 0x0D 0x0D (carriage return, carriage return). If you would like a carriage return not to be sent you could set the **Convert From Char B** property to 0x0D (carriage return) and set the **Convert To String B** property to no string.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Convert From Char B property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	02	1C	FF

Example Set Convert From Char B property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Convert From Char B property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	1C

Example Convert From Char B property Response (Hex):

Result Code	Data Len	Prp Value
00	01	FF

5.2.10 Property 0x1D - Convert to String B

Property ID:	29 (0x1D)
Property Type:	String
Length:	0 – 7 bytes
Get Property:	Yes
Set Property:	Yes
Default Value:	The default value is no string with a length of zero.

The value is an ASCII string that represents the device's **Convert To String B** property. This string can be 0 – 7 bytes long. This string is sent in place of the character specified in the **Convert From Char B** property. See the **Convert From Char B** property for more information and examples.

This property is stored in non-volatile memory, so it will persist when the device is power cycled. When this property is changed, the device must be reset (see Command Number 2) or power cycled for these changes to take effect.

Example Set Convert To String B property Request (Hex):

Cmd Num	Data Len	Prp ID	Prp Value
01	03	1D	0D 0D

Example Set Convert To String B property Response (Hex):

Result Code	Data Len	Data
00	00	

Example Get Convert To String B property Request (Hex):

Cmd Num	Data Len	Prp ID
00	01	1D

Example Get Convert To String B property Response (Hex):

Result Code	Data Len	Prp Value
00	02	0D 0D

6 Encryption

6.1 About Encrypted MSR Data

The UART/RS-232 MagneSafe IntelliHead^{TM} Reader is a compact magnetic stripe card reader that conforms to ISO standards. In addition to reading multiple tracks of data from a card, this reader also includes MagnePrint technology. The MagnePrint data will be included with the track data on each transaction. In order to maximize card security, this reader can incorporate data encryption to protect the card contents and the MagnePrint information. The serial output will be encrypted before it leaves the encapsulated head.

A separate LED (Light Emitting Diode) indicator provides the operator with continuous status of the operation of each Reader (MICR and MSR) module incorporated into the device.

When a card is swiped through the Reader, the track data and MagnePrint information may, depending on how MICRSafe is configured, be TDEA (Triple Data Encryption Algorithm, aka, Triple DES) encrypted using DUKPT (Derived Unique Key Per Transaction) key management. This method of key management uses a base derivation key to encrypt a key serial number that produces an initial encryption key which is injected into the Reader prior to deployment. After each transaction, the encryption key is modified per the DUKPT algorithm so that each transaction uses a unique key. Thus, the data will be encrypted with a different encryption key for each transaction. More detailed information about the encryption methods used by this device can be found in the MagneSafe MSR Technical Manual, P/N D99875433. If the encryption feature is not set, then MICRSafe will output data in SureSwipe format. See P/N D99875206 for more information about this non-encrypted output format.

6.2 About Encrypted Check Data

MICRSafe can be similarly configured to encrypt MICR data from scanned checks. See **Table 6-1** for information about how the host should interpret this data.

Table 6-1 - Interpreting Encrypted MICR Data

Length	Data
1 byte	Response Code (0x00 = OK)
1 byte	Data Length of all data below this byte
16 bytes	Device Serial Number (DSN)
1 byte	Anticipated length of Encrypted Data Block after decryption
10 bytes	Key Serial Number (KSN) of encryption key used to encrypt the data
Variable, multiple of 8	Encrypted Data Block
4 bytes	Message Authentication Code (MAC)

For example:
`003F4233384335363930313131313741410001E9013930B38C5690000172C699D5D7B14AB37B6DD49AB9C41B75976FF2B340C7EFE6575B5E6CE38BB98A896A9ACA6` would be interpreted as:

Response code OK: 00
Data Length: 3F
Device Serial Number: 4233384335363930313131313137414100
Anticipated length of decrypted data: 1E
Key Serial Number: 9013930B38C569000017
Encrypted Data Block:
2C699D5D7B14AB37B6DD49AB9C41B75976FF2B340C7EFE6575B5E6CE38BB98A8
Message Authentication Code: 96A9ACA6

The host would use the KSN to derive the appropriate DUKPT key, decrypt the encrypted data block, and check it against its anticipated length. It may also use the included MAC to verify the message originated from an authentic source.

Appendix A Technical Specifications

MICRSafe Technical Specifications	
Reference Standards and Certifications	
ISO/CDL/AAMVA	
Physical Characteristics	
Dimensions (L x W x H):	6.25 in. L x 4.0 in. W x 4.25 in. H (159mm x 102mm x 108mm)
Cable Length:	6 ft.
Weight	3.0 lbs., MSR and adapter included
Card Read Characteristics	
Magnetic Stripe Reader:	Tracks 1, 2, and 3
Check Read Characteristics	
Check Read / Decode / Transit Time:	2 seconds
MICR fonts supported	E13-B CMC-7
Check Read / Decode / Transit Time:	2 seconds
Electrical Characteristics	
Power Inputs:	120 VAC, 50/60 Hz
Wired Connection Types:	USB Type B Mini DIN-9 male
Output Signal Levels:	12 VDC, 200 mA (Idle), 600mA (Operating)
Environmental Tolerance	
Ingress Protection:	Not Applicable
Operating Temperature:	32°F to 122°F (0°C to 50°C)
Operating Relative Humidity:	10% to 90% non-condensing
Storage Temperature:	-22°F to 140°F (-30°C to 60°C)
Storage Relative Humidity:	Up to 95% non-condensing

Appendix B Format List

For check reading, MICRSafe provides the flexibility to format the MICR fields and build a specific output string that will be transmitted to the host. These output strings are referred to as formats. The Reader has a built-in list of formats (described below) from which the user may select one to become the active format every time a check is read. The formats may be selected using the FC command (see section **4 Legacy Commands**) or Insta-Change checks provided by MagTek.

Each format is assigned a 4-digit number which indicates the format number. ***Note:*** *The formats listed in this section apply only to U.S. and Canadian checks. The MICR line on checks from other countries will not be broken or parsed as described in these formats.*

A complete description for each format follows.

Fmt 00xx: Raw Data Format - sends the entire MICR line - where:

xx	Transit	On-Us	Amount	Dash	Error
00	T	U	$	-	?
01	t	o	a	d	?
02	T	O	A	D	?

Fmt 0800: [transit] [acct #]

```
• [transit]:      - all characters in the field
                  - keep dashes

• [acct #]:    - all characters are sent
                  - remove spaces and dashes
```

Fmt 1100: [transit] 'T' [acct #] 'A' [check #]

```
• [transit]:      - all characters in the field
                  - keep dashes

• [acct #]:    - all characters are sent
                  - remove spaces and dashes

• [check #]:      - all characters in the field
```

Fmt 1200: [transit] 'T' [acct #] 'A' [check #]

```
• [transit]:      - all characters in the field
                  - keep dashes

• [acct #]:    - all characters are sent
                  - remove spaces and dashes

• [check #]:      - always 6 characters, zero filled
```

Fmt 2400: [transit] 'T' [acct #] 'A' [check #] 'C' [amount] '$'

```
• [transit]:      - all characters in the field
                  - keep dashes

• [acct #]:    - all characters are sent
                  - remove spaces and dashes

• [check #]:      - always 6 characters, zero filled

• [amount]:    - all characters in the field
```

Fmt 3100: [transit] '/' [acct #] '/' [check #]

```
• [transit]:      - all characters in the field
                  - remove dashes

• [acct #]:    - all characters are sent
                  - remove spaces and dashes

• [check #]:      - maximum of 10 digits
                  - remove spaces and dashes
                  - if no check number, remove preceding slash ('/')
```

Fmt 3800: 'T' [transit] 'A' [acct #] 'C' [check #]

```
• [transit]:      - all characters in the field
                  - keep dashes
• [acct #]:    - all characters are sent
```

- include leading characters
- keep spaces and dashes

- **[check #]:** -all characters in the field

Fmt 4500: **[transit] <CR> [acct #] <CR> [check #]**

- **[transit]:** - all characters in the field
  - remove dashes
- **[acct #]:** - all characters are sent
  - remove spaces, dashes and leading zeros
- **[check #]:** - all characters in the field

Fmt 6500: **'!' [transit] '/' [acct #] '/' [check #] '/' [amount]**

- **[transit]:** - all characters in the field
  - remove dashes
- **[acct #]:** - all characters are sent
  - remove spaces and dashes
- **[check #]:** - all characters in the field
  - remove dashes and spaces
- **[amount]:** - all characters in the field
  - remove dashes and spaces

Fmt 7600: **'T' [transit] 'A' [acct #] 'C' [check #] 'M' [raw data]**

- **[transit]:** - all characters in the field
  - remove dashes and spaces
- **[acct #]:** - all characters are sent
  - remove dashes and spaces
- **[check #]:** - all characters in the field- remove dashes and spaces
- **[raw data]:** - translate MICR symbols to t,o,a,d

Fmt 7700: The Flexible Format

Select this format to activate a preloaded Flexible Format. The Flexible Format is a feature that allows the user to create custom MICR formats. The Flexible formats can be easily created and downloaded using the Windows based MICRbase program provided by MagTek (P/N 22000021). For more detailed information refer to MICRbase reference manual ***D99875102***.

Appendix C Check Reading

The characters printed on the bottom line of commercial and personal checks are special. They are printed with magnetic ink to meet specific standards. These characters can be read by a MICRSafe at higher speeds and with more accuracy than manual data entry. Two MICR character sets are used worldwide; they are: E13-B and CMC-7. The E13-B set is used in the US, Canada, Australia, United Kingdom, Japan, India, Mexico, Venezuela, Colombia, and the Far East. The CMC-7 set is used in France, Spain, other Mediterranean countries, and most South American countries.

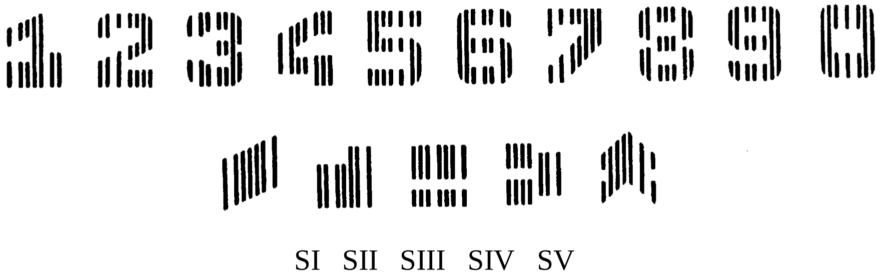
C.1 E13-B Character Set

The MICR font character set E13-B includes digits 0 through 9 and four symbols. The numbers found on U.S. checks are of the E13-B character set. The numbers and symbols of E13-B are as follows:

1	6		
2	7	⑆	Transit symbol
3	8	⑈	Dash Symbol
4	9	⑉	On-Us Symbol
5	0	⑇	Amount Symbol

C.2 CMC-7 Character Set

The numbers and symbols of the CMC-7 character set are as follows:



The nonnumeric CMC-7 characters are translated by MICRSafe as shown in **Table 6-2**.

Table 6-2 - CMC-7 Non-numeric Characters

CMC-7 Character	MICRSafe Output
SI	A
SII	B
SIII	C
SIV	D
SV	E

C.3 Check Layouts

Personal checks with MICR fields are shown in **Figure 6-1**. Business checks are shown in **Figure 6-2**. The digits 1 through 4 in the illustrations are described below under MICR Fields.

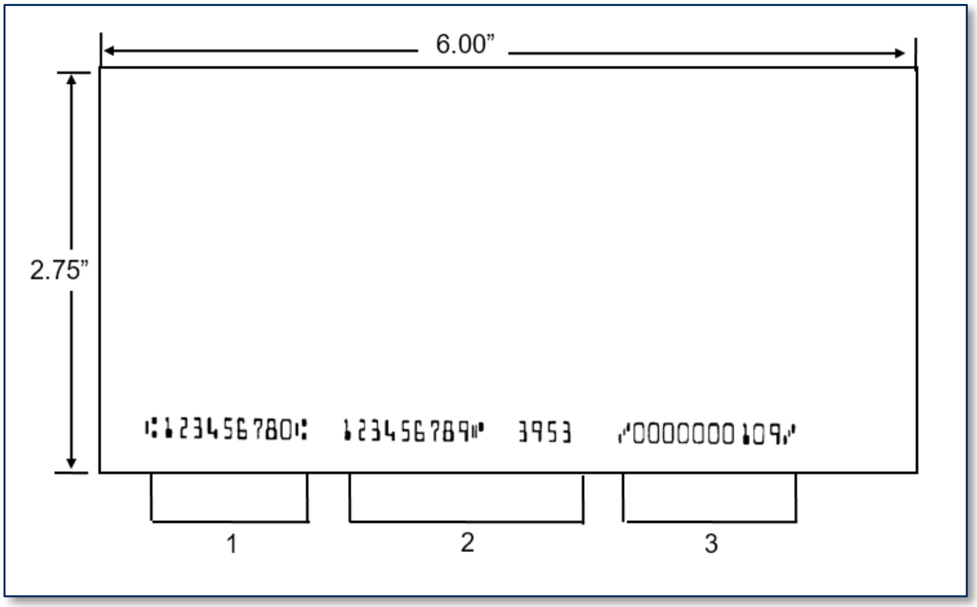


Figure 6-1 -Personal Checks

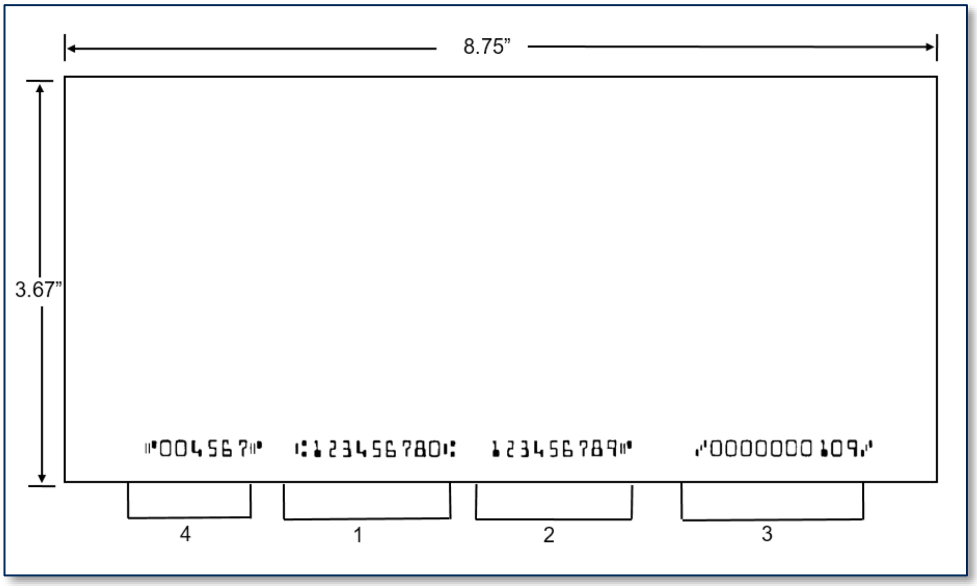


Figure 6-2 -Business Checks

C.4 MICR Fields

The numbers 1 through 4 refer to the numbers below the checks on the illustration and represent the 4 MICR fields.

C.4.1 1-Transit Field

The Transit field is a 9-digit field bracketed by two Transit symbols. The field is subdivided as follows:

- Digits 1-4 Federal Reserve Routing Number
- Digits 5-8 Bank ID Number (American Banking Association)
- Digit 9 Check Digit

C.4.2 2-On-Us Field

The On-Us field is variable, up to 19 characters (including symbols). Valid characters are digits, spaces, dashes, and On-Us symbols. The On-Us field contains the account number and may also contain a serial number (Check number) and/or a transaction code. Note that an On-Us symbol must always appear to the right of the account number.

C.4.3 3-Amount Field

The Amount field is a 10-digit field bracketed by Amount symbols. The field is always zero-filled to the left.

C.4.4 4-Auxiliary On-Us Field

The Auxiliary On-Us field is variable, 4-10 digits, bracketed by two On-Us symbols. This field is not present on personal checks. On business checks, this field contains the check serial number.

Appendix D Troubleshooting Guide

D.1 Requirements

- Personal Computer.
- USB Cable, P/N 22553301
- DC adapter, P/N 64300118 (64300121 for international customers).
- MICRbase setup program, P/N 22000021.
- Sample checks, P/N 96530005.
- A small bottle of compressed air.
- A cleaning card, P/N 96700006.

D.2 Setup

- Plug the USB Cable, P/N 22553301, into MICRSafe and the host computer.
- Power on the MICR Reader.
- Run the MICRbase program on the PC.
- Start troubleshooting procedure at **Step 00: Check LED**.

D.3 Step 00: Check LED

Check the status of the LED indicator:

- Off: Continue to **Step 01: Check the Power to MICRSafe**.
- Green: Continue to **Step 02: Read a** Check.
- Blinking red: Continue to **Step 10: Motor Sensor is Blocked**.
- Blinking red/green: Continue to **Step 11: EMF Noise / Interference**.
- Red or Orange: Continue to **Step 17: Return MICRSafe for Service**.

D.4 Step 01: Check the Power to MICRSafe

Possible causes for this problem are:

- DC adapter connection to outlet - make sure the DC adapter is securely connected to outlet on the wall or power strip.
- DC adapter connection to MICRSafe - make sure the DC adapter is securely connected to MICRSafe’s power connector.
- Power strip - if using a power strip, make sure the strip is connected to outlet on the wall and the switch on the strip is turned on.
- DC adapter is defective - replace the DC adapter.

D.5 Step 02: Read a Check

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 00: Check LED**.
- If no, continue to **Step 17: Return MICRSafe for Service**.

Read a check through MICRSafe:

- If the check is transported all the way around the check path, continue to **Step 03: Did PC Receive Data?**.
- If the check gets stuck in the check path, continue to **Step 09: Path is Obstructed**.

- If the motor does not turn on, continue to **Step 17: Return MICRSafe for Service**.

D.6 Step 03: Did PC Receive Data?

After the check is read, did the PC receive any data?

- If yes, continue to **Step 04: Analyze Data**.
- If no, continue to **Step 05: Verify parameters**.

D.7 Step 04: Analyze Data

Analyze the data received by the PC:

◊ If the data is good, continue to **Step 15: No Problem Found**.
◊ If the data contains one or more '?', continue to **Step 06: Read Error**.
◊ If the data is missing characters, continue to **Step 07: Missing Characters**.
◊ If the data is good but not what is expected, continue to **Step 08: Incorrect Format**.

D.8 Step 05: Verify parameters

Use the "Configure" option within MICRbase to verify the following parameters:

- "Send Data After Error" - if this option is set to NO, MICRSafe will not send any data after a read error. Use MICRbase to change this option to YES.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.
- If no, continue to **Step 13: No MICR Data Detected**.

D.9 Step 06: Read Error

Possible causes for this problem are:

- Interference - MICRSafe may be too close to a monitor, DC adapter or magnetic device. Move MICRSafe away from the source of interference.
- Printing problem - the check being read may not meet the requirements of the ANSI Standards. Use one of the sample checks provided by MagTek.
- Feeding the check - do not hold on to the check as it goes around the path. Release the check immediately after MICRSafe engages it. Also, make sure that the front end is not tilted up while the check is being read.
- Foreign debris – power off MICRSafe and try to push out any loose debris on the check path. Use a cleaning card and force it through the check path (this is a manual process, the motor will not turn on). Try this procedure several times until the debris comes out. Power on MICRSafe again.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.
- If no, continue to **Step 13: No MICR Data Detected**.

D.10 Step 07: Missing Characters

Possible causes for this problem are:

- Feeding the check - When feeding the check, make sure that the MICR line is at the bottom and the printed side of the check is facing the MagTek logo on the MICR Reader.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.

- If no, continue to **Step 08: Incorrect Format**.

D.11 Step 08: Incorrect Format

Possible causes for this problem are:

- Incorrect Format Number - the current Check data format in MICRSafe is not the desired format. Use the “Configure” option within MICRbase to verify/change the format.
- Incorrect Message Format - the current Message format in MICRSafe is not the desired format. Use the “Configure” option within MICRbase to verify/change the Message format.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.
- If no, continue to **Step 17: Return MICRSafe for Service**.

D.12 Step 09: Path is Obstructed

Foreign debris is obstructing the check path:

- Loose debris - power off MICRSafe and try to push out any loose debris on the check path. Use a cleaning card and force it through the check path (this is a manual process, the motor will not turn on). Try this procedure several times until the debris comes out. Power on the MICR Reader.
- Wedged debris - the debris is wedged in and cannot be removed with the procedure described above.

Is the foreign debris removable?

- If yes, remove and continue to **Step 02: Read a Check**.
- If no, continue to **Step 17: Return MICRSafe for Service**.

D.13 Step 10: Motor Sensor is Blocked

The Motor sensor may be blocked by dust build-up or foreign debris (see **Figure 6-3**). Use forced air to clean the sensor.

Power off MICRSafe and then power on again; then observe the LED indicator:

- If the LED indicator blinks red, continue to **Step 17: Return MICRSafe for Service**.
- Any other LED indicator status, continue to **Step 00: Check LED**.

D.14 Step 11: EMF Noise / Interference

When idle, if EMF detect is set to YES (see HW Command, section **4 Legacy Commands**), MICRSafe monitors the signal coming from the MICR head. If any signal (noise/interference) with an amplitude large enough to affect check reading is detected, the LED indicator blinks red/green. Possible sources of EMF are monitors, DC adapters, or magnetic devices. Set EMF to NO or move MICRSafe at least 6 inches away from the source of noise/interference.

Determine if the condition described above is true:

- If yes, rectify and continue to **Step 00: Check LED**.
- If no, continue to **Step 12: Data Sensor is Blocked**.

D.15 Step 12: Data Sensor is Blocked

The data sensor may be blocked (see **Figure 6-3**). Try one or both of the following procedures:

- Forced air - use forced air to clean the sensor.

- Cleaning card - power off MICRSafe and try to push out any loose debris in the check path. Use a cleaning card and force it through the check path (this is a manual process, the motor will not turn on). Try this procedure several times until the debris comes out.

Power off MICRSafe, then power on again and observe the LED indicator:

- If the LED indicator blinks red/green, continue to **Step 17: Return MICRSafe for Service**.
- Any other LED indicator status, continue to **Step 00: Check LED**.

D.16 Step 13: No MICR Data Detected

Possible causes for this problem are:

- No MICR characters - the ink used to print the MICR characters does not have magnetic properties. Try one of the sample checks provided by MagTek.
- Feeding the check - when feeding the check, make sure that the MICR line is at the bottom and the printed side of the check is facing the MagTek logo on the MICR Reader.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.
- If no, continue to **Step 14: Cable Problem**.

D.17 Step 14: Cable Problem

Possible causes for this problem are:

- Loose connection - the cable connector on the PC or MICRSafe may be loose. Make sure that both connectors are tightly connected.
- Damaged cable - the connectors, pins or wires in the cable may be damaged. Replace cable.

Determine if any of the conditions described above are true:

- If yes, rectify and continue to **Step 02: Read a Check**.
- If no, continue to **Step 17: Return MICRSafe for Service**.

D.18 Step 15: No Problem Found

MICRSafe is operating properly. If you have additional concerns or requirements please contact your MagTek representative.

## D.19 Step 16	Read Insta-Change Check

Read Insta-Change check with the appropriate settings. Return to **Step 00: Check LED**. If condition persists, continue to **Step 17: Return MICRSafe for Service**.

D.20 Step 17: Return MICRSafe for Service

MICRSafe has a problem that needs further analysis, testing, and possibly repair. Please contact your supplier or MagTek Support Services at (651) 415-6800, for additional troubleshooting and (if necessary) repair. Please have the device available and ready prior to contacting your supplier or MagTek.

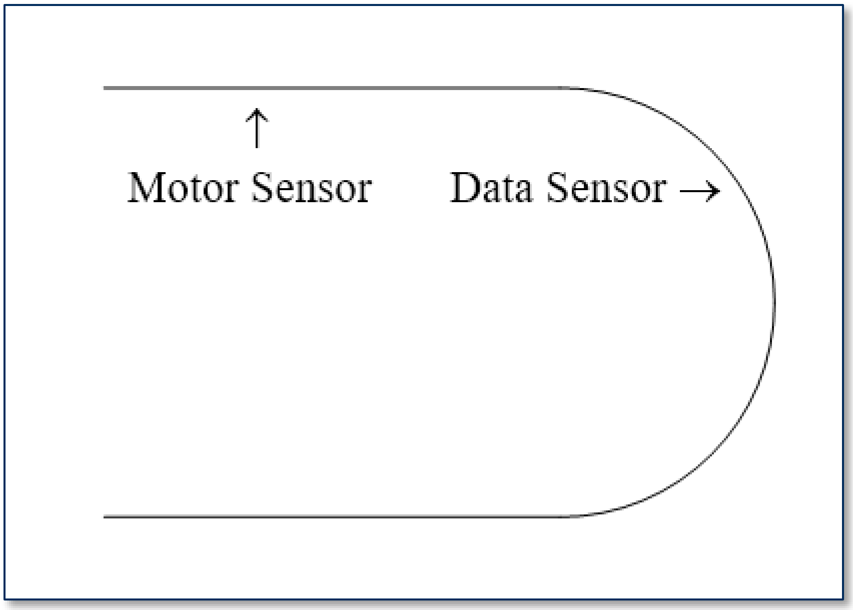


Figure 6-3 - Sensor Location

Appendix E ASCII Codes

This table lists the ASCII (American Standard Code for Information Interchange) codes. ASCII is a 7-bit code which is represented here by a pair of hexadecimal digits.

ASCII	Hex	Dec	ASCII	Hex	Dec	ASCII	Hex	Dec	ASCII	Hex	Dec
NUL	00	0	SP	20	32	@	40	64	`	60	96
SOH	01	1	!	21	33	A	41	65	a	61	97
STX	02	2	"	22	34	B	42	66	b	62	98
ETX	03	3	#	23	35	C	43	67	c	63	99
EOT	04	4	$	24	36	D	44	68	d	64	100
ENQ	05	5	%	25	37	E	45	69	e	65	101
ACK	06	6	&	26	38	F	46	70	f	66	102
BEL	07	7	'	27	39	G	47	71	g	67	103
BS	08	8	(	28	40	H	48	72	h	68	104
HT	09	9	)	29	41	I	49	73	i	69	105
LF	0A	10	*	2A	42	J	4A	74	j	6A	106
VT	0B	11	+	2B	43	K	4B	75	k	6B	107
FF	0C	12	,	2C	44	L	4C	76	l	6C	108
CR	0D	13	-	2D	45	M	4D	77	m	6D	109
SO	0E	14	.	2E	46	N	4E	78	n	6E	110
SI	0F	15	/	2F	47	O	4F	79	o	6F	111
DLE	10	16	0	30	48	P	50	80	p	70	112
DC1	11	17	1	31	49	Q	51	81	q	71	113
DC2	12	18	2	32	50	R	52	82	r	72	114
DC3	13	19	3	33	51	S	53	83	s	73	115
DC4	14	20	4	34	52	T	54	84	t	74	116
NAK	15	21	5	35	53	U	55	85	u	75	117
SYN	16	22	6	36	54	V	56	86	v	76	118
ETB	17	23	7	37	55	W	57	87	w	77	119
CAN	18	24	8	38	56	X	58	88	x	78	120
EM	19	25	9	39	57	Y	59	89	y	79	121
SUB	1A	26	:	3A	58	Z	5A	90	z	7A	122
ESC	1B	27	;	3B	59	[	5B	91	{	7B	123
FS	1C	28	<	3C	60	\	5C	92	|	7C	124
GS	1D	29	=	3D	61	]	5D	93	}	7D	125

ASCII	Hex	Dec	ASCII	Hex	Dec	ASCII	Hex	Dec	ASCII	Hex	Dec
RS	1E	30	>	3E	62	^	5E	94	~	7E	126
US	1F	31	?	3F	63	_	5F	95	DEL	7F	127

Appendix F Usage ID Definitions

This appendix is from the following document found on www.usb.org***: Universal Serial Bus HID Usage Tables, Version 1.12*** and specifically for this manual, ***Section 10, Keyboard/Keypad Page (0x07)***.

Keyboard/Keypad Page (0x07)
This section is the Usage Page for key codes to be used in implementing a USB keyboard. A Boot Keyboard (84-, 101- or 104-key) should at a minimum support all associated usage codes as indicated in the "Boot" column below.

The usage type of all key codes is Selectors (Sel), except for the modifier keys Keyboard Left Control (0x224) to Keyboard Right GUI (0x231) which are Dynamic Flags (DV).

Note
A general note on Usages and languages: Due to the variation of keyboards from language to language, it is not feasible to specify exact key mappings for every language. Where this list is not specific for a key function in a language, the closest equivalent key position should be used, so that a keyboard may be modified for a different language by simply printing different keycaps. One example is the Y key on a North American keyboard. In Germany this is typically Z. Rather than changing the keyboard firmware to put the Z Usage into that place in the descriptor list, the vendor should use the Y Usage on both the North American and German keyboards. This continues to be the existing practice in the industry, in order to minimize the number of changes to the electronics to accommodate other languages.

Table 6-3 - Keyboard / Keypad

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
0	00	Reserved (no event indicated) 9	N/A	√	√	√	4/101/104
1	01	Keyboard ErrorRollOver9	N/A	√	√	√	4/101/104
2	02	Keyboard POSTFail9	N/A	√	√	√	4/101/104
3	03	Keyboard ErrorUndefined9	N/A	√	√	√	4/101/104
4	04	Keyboard a and A4	31	√	√	√	4/101/104
5	05	Keyboard b and B	50	√	√	√	4/101/104
6	06	Keyboard c and C4	48	√	√	√	4/101/104
7	07	Keyboard d and D	33	√	√	√	4/101/104
8	08	Keyboard e and E	19	√	√	√	4/101/104
9	09	Keyboard f and F	34	√	√	√	4/101/104
10	0A	Keyboard g and G	35	√	√	√	4/101/104
11	0B	Keyboard h and H	36	√	√	√	4/101/104
12	0C	Keyboard i and I	24	√	√	√	4/101/104
13	0D	Keyboard j and J	37	√	√	√	4/101/104

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
14	0E	Keyboard k and K	38	√	√	√	4/101/104
15	0F	Keyboard l and L	39	√	√	√	4/101/104
16	10	Keyboard m and M	52	√	√	√	4/101/104
17	11	Keyboard n and N	51	√	√	√	4/101/104
18	12	Keyboard o and O4	25	√	√	√	4/101/104
19	13	Keyboard p and P4	26	√	√	√	4/101/104
20	14	Keyboard q and Q4	27	√	√	√	4/101/104
21	15	Keyboard r and R	20	√	√	√	4/101/104
22	16	Keyboard s and S4	32	√	√	√	4/101/104
23	17	Keyboard t and T	21	√	√	√	4/101/104
24	18	Keyboard u and U	23	√	√	√	4/101/104
25	19	Keyboard v and V	49	√	√	√	4/101/104
26	1A	Keyboard w and W4	18	√	√	√	4/101/104
27	1B	Keyboard x and X4	47	√	√	√	4/101/104
28	1C	Keyboard y and Y4	22	√	√	√	4/101/104
29	1D	Keyboard z and Z4	46	√	√	√	4/101/104
30	1E	Keyboard 1 and !4	2	√	√	√	4/101/104
31	1F	Keyboard 2 and !4	3	√	√	√	4/101/104
32	20	Keyboard 3 and #4	4	√	√	√	4/101/104
33	21	Keyboard 4 and $4	5	√	√	√	4/101/104
34	22	Keyboard 5 and %4	6	√	√	√	4/101/104
35	23	Keyboard 6 and ^4	7	√	√	√	4/101/104
36	24	Keyboard 7 and &4	8	√	√	√	4/101/104
37	25	Keyboard 8 and *4	9	√	√	√	4/101/104
38	26	Keyboard 9 and (4	10	√	√	√	4/101/104
39	27	Keyboard 0 and)4	11	√	√	√	4/101/104
40	28	Keyboard Return (ENTER)5	43	√	√	√	4/101/104
41	29	Keyboard ESCAPE	110	√	√	√	4/101/104
42	2A	Keyboard DELETE (Backspace)	15	√	√	√	4/101/104
43	2B	Keyboard Tab	16	√	√	√	4/101/104

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
44	2C	Keyboard Spacebar	61	√	√	√	4/101/104
45	2D	Keyboard - and (underscore)4	12	√	√	√	4/101/104
46	2E	Keyboard = and +4	13	√	√	√	4/101/104
47	2F	Keyboard [and {4	27	√	√	√	4/101/104
48	30	Keyboard] and }4	28	√	√	√	4/101/104
49	31	Keyboard \ and |	29	√	√	√	4/101/104
50	32	Keyboard Non-US # and ~2	42	√	√	√	4/101/104
51	33	Keyboard ; and :4	40	√	√	√	4/101/104
52	34	Keyboard ‘ and “4	41	√	√	√	4/101/104
53	35	Keyboard Grave Accent and Tilde4	1	√	√	√	4/101/104
54	36	Keyboard , and <4	53	√	√	√	4/101/104
55	37	Keyboard . and >4	54	√	√	√	4/101/104
56	38	Keyboard / and ?	55	√	√	√	4/101/104
57	39	Keyboard Caps Lock11	30	√	√	√	4/101/104
58	3A	Keyboard F1	112	√	√	√	4/101/104
59	3B	Keyboard F2	113	√	√	√	4/101/104
60	3C	Keyboard F3	114	√	√	√	4/101/104
61	3D	Keyboard F4	115	√	√	√	4/101/104
62	3E	Keyboard F5	116	√	√	√	4/101/104
63	3F	Keyboard F6	117	√	√	√	4/101/104
64	40	Keyboard F7	118	√	√	√	4/101/104
65	41	Keyboard F8	119	√	√	√	4/101/104
66	42	Keyboard F9	120	√	√	√	4/101/104
67	43	Keyboard F10	121	√	√	√	4/101/104
68	44	Keyboard F11	122	√	√	√	101/104
69	45	Keyboard F12	123	√	√	√	101/104
70	46	Keyboard PrintScreen1	124	√	√	√	101/104
71	47	Keyboard Scroll Lock11	125	√	√	√	4/101/104
72	48	Keyboard Pause1	126	√	√	√	101/104
73	49	Keyboard Insert1	75	√	√	√	101/104

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
74	4A	Keyboard Home1	80	√	√	√	101/104
75	4B	Keyboard PageUp1	85	√	√	√	101/104
76	4C	Keyboard Delete Forward1;14	76	√	√	√	101/104
77	4D	Keyboard End1	81	√	√	√	101/104
78	4E	Keyboard PageDown1	86	√	√	√	101/104
79	4F	Keyboard RightArrow1	89	√	√	√	101/104
80	50	Keyboard LeftArrow1	79	√	√	√	101/104
81	51	Keyboard DownArrow1	84	√	√	√	101/104
82	52	Keyboard UpArrow1	83	√	√	√	101/104
83	53	Keypad Num Lock and Clear11	90	√	√	√	101/104
84	54	Keypad /1	95	√	√	√	101/104
85	55	Keypad *	100	√	√	√	4/101/104
86	56	Keypad -	105	√	√	√	4/101/104
87	57	Keypad +	106	√	√	√	4/101/104
88	58	Keypad ENTER5	108	√	√	√	101/104
89	59	Keypad 1 and End	93	√	√	√	4/101/104
90	5A	Keypad 2 and Down Arrow	98	√	√	√	4/101/104
91	5B	Keypad 3 and PageDn	103	√	√	√	4/101/104
92	5C	Keypad 4 and Left Arrow	92	√	√	√	4/101/104
93	5D	Keypad 4 and Left Arrow	97	√	√	√	4/101/104
94	5E	Keypad 4 and Left Arrow	102	√	√	√	4/101/104
95	5F	Keypad 7 and Home	91	√	√	√	4/101/104
96	60	Keypad 8 and Up Arrow	96	√	√	√	4/101/104
97	61	Keypad 9 and PageUp	101	√	√	√	4/101/104
98	62	Keypad 0 and Insert	99	√	√	√	4/101/104
99	63	Keypad . and Delete	104	√	√	√	4/101/104
100	64	Keyboard Non-US \ and |3;6	45	√	√	√	4/101/104
101	65	Keyboard Application10	129	√		√	104
102	66	Keyboard Power9 =			√	√	
103	67	Keypad =			√		

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
104	68	Keyboard F13	62		√		
105	69	Keyboard F14	63		√		
106	6A	Keyboard F15	64		√		
107	6B	Keyboard F16	65				
107	6C	Keyboard F17					
109	6D	Keyboard F18					
110	6E	Keyboard F19					
111	6F	Keyboard F20					
112	70	Keyboard F21					
113	71	Keyboard F22					
114	72	Keyboard F23					
115	73	Keyboard F24					
116	74	Keyboard Execute				√	
117	75	Keyboard Help				√	
118	76	Keyboard Menu				√	
119	77	Keyboard Select				√	
120	78	Keyboard Stop				√	
121	79	Keyboard Again				√	
122	7A	Keyboard Undo				√	
123	7B	Keyboard Cut				√	
124	7C	Keyboard Copy				√	
125	7D	Keyboard Paste				√	
126	7E	Keyboard Find				√	
127	7F	Keyboard Mute				√	
128	80	Keyboard Volume Up				√	
129	81	Keyboard Volume Down				√	
130	82	Keyboard Locking Caps Lock12				√	
131	83	Keyboard Locking Num Lock12				√	
132	84	Keyboard Locking Scroll Lock12				√	
133	85	Keypad Comma27	107				

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
134	86	Keypad Equal Sign29					
135	87	Keyboard International115-28	56				
136	88	Keyboard International216					
137	89	Keyboard International317					
138	8A	Keyboard International418					
139	8B	Keyboard International519					
140	8C	Keyboard International620					
141	8D	Keyboard International721					
142	8E	Keyboard International822					
143	8F	Keyboard International922					
144	90	Keyboard Lang125					
145	91	Keyboard Lang226					
146	92	Keyboard Lang330					
147	93	Keyboard Lang431					
148	94	Keyboard Lang532					
149	95	Keyboard Lang68					
150	96	Keyboard Lang78					
151	97	Keyboard Lang88					
152	98	Keyboard Lang98					
153	99	Keyboard Alternate Erase7					
154	9A	Keyboard Sys/Req Attention1					
155	9B	Keyboard Cancel					
156	9C	Keyboard Clear					
157	9D	Keyboard Prior					
158	9E	Keyboard Return					
159	9F	Keyboard Separator					
160	A0	Keyboard Out					
161	A1	Keyboard Oper					
162	A2	Keyboard Clear/Again					
163	A3	Keyboard Cr/Sel/Props					

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
164	A4	Keyboard Ex Sel					
165-175	A5-CF	Reserved					
176	B0	Keypad 00					
177	B1	Keypad 000					
178	B2	Thousands Separator33					
179	B3	Decimal Separator33					
180	B4	Currency Unit34					
181	B5	Currency Sub-unit34					
182	B6	Keypad (					
183	B7	Keypad)					
184	B8	Keypad {					
185	B9	Keypad}					
186	BA	Keypad Tab					
187	BB	Keypad Backspace					
188	BC	Keypad A					
189	BD	Keypad B					
190	BE	Keypad C					
191	BF	Keypad D					
192	C0	Keypad E					
193	C1	Keypad F					
194	C2	Keypad XOR					
195	C3	Keypad ^					
196	C4	Keypad %					
197	C5	Keypad <					
198	C6	Keypad >					
199	C7	Keypad &					
200	C8	Keypad &&					
201	C9	Keypad |					
202	CA	Keypad ||					
203	CB	Keypad :					

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
204	CC	Keypad #					
205	CD	Keypad Space					
206	CE	Keypad @					
207	CF	Keypad !					
208	D0	Keypad Memory Store					
209	D1	Keypad Memory Recall					
210	D2	Keypad Memory Clear					
211	D3	Keypad Memory Add					
212	D4	Keypad Memory Subtract					
213	D5	Keypad Memory Multiple					
214	D6	Keypad Memory Divide					
215	D7	Keypad +/-					
216	D8	Keypad Clear					
217	D9	Keypad Clear Entry					
218	DA	Keypad Binary					
219	DB	Keypad Octal					
220	DC	Keypad Decimal					
221	DD	Keypad Hexadecimal					
222-223	DE-DF	Reserved					
224	E0	Keyboard LeftControl	58	√	√	√	
225	E1	Keyboard LeftShift	44	√	√	√	
226	E2	Keyboard LeftA;t	60	√	√	√	
227	E3	Keyboard Left GUI10;23	127	√	√	√	
228	E4	Keyboard RightControl	64	√	√	√	
229	E5	Keyboard RightShift	57	√	√	√	
230	E6	Keyboard RightAlt	62	√	√	√	
231	E7	Keyboard Right GUI10;24	128	√	√	√	
232 – 65535	E8-FFFF	Reserved					

1) Usage of keys is not modified by the state of the Control, Alt, Shift or Num Lock keys. That is, a key does not send extra codes to compensate for the state of any Control, Alt, Shift or Num Lock keys.
2) Typical language mappings: US: \| Belg: µ`£ FrCa: <}> Dan:’* Dutch: <> Fren:*µ Ger: #’ Ital: ù§ LatAm: }`] Nor:,* Span: }Ç Swed: ,* Swiss: $£ UK: #~.
3) Typical language mappings: Belg:<\> FrCa:«°» Dan:<\> Dutch:]|[Fren:<> Ger:<|> Ital:<> LatAm:<> Nor:<> Span:<> Swed:<|> Swiss:<\> UK:\| Brazil: \|.
4) Typically remapped for other languages in the host system.
5) Keyboard Enter and Keypad Enter generate different Usage codes.
6) Typically near the Left-Shift key in AT-102 implementations.
7) Example, Erase-Eaze™ key.
8) Reserved for language-specific functions, such as Front End Processors and Input Method Editors.
9) Reserved for typical keyboard status or keyboard errors. Sent as a member of the keyboard array. Not a physical key.
10) Windows key for Windows 95, and “Compose.”
11) Implemented as a non-locking key; sent as member of an array.
12) Implemented as a locking key; sent as a toggle button. Available for legacy support; however, most systems should use the non-locking version of this key.
13) Backs up the cursor one position, deleting a character as it goes.
14) Deletes one character without changing position.
15) See additional footnotes in Universal Serial Bus HID Usage Tables, Copyright © 1996-2005, USB Implementers Forum.
16) See above.
17) See above.
18) See above.
19) See above.
20) See above.
21) Toggle Double-Byte/Single-Byte mode.
22) Undefined, available for other Front End Language Processors.
23) Windowing environment key, for example Microsoft Left Win key, Mac Left Apple key, Sun Left Meta key
24) Windowing environment key, examples are Microsoft® RIGHT WIN key, Macintosh® RIGHT APPLE key, Sun® RIGHT META key.
25) Hangul/English toggle key. This usage is used as an input method editor control key on a Korean language keyboard.
26) Hanja conversion key. This usage is used as an input method editor control key on a Korean language keyboard.
27) Keypad Comma is the appropriate usage for the Brazilian keypad period (.) key. This represents the closest possible match, and system software should do the correct mapping based on the current locale setting.
28) Keyboard International1 should be identified via footnote as the appropriate usage for the Brazilian forward-slash (/) and question-mark (?) key. This usage should also be renamed to either "Keyboard Non-US / and ?" or to "Keyboard International1" now that it's become clear that it does not only apply to Kanji keyboards anymore.
29) Used on AS/400 keyboards.
30) Defines the Katakana key for Japanese USB word-processing keyboards.
31) Defines the Hiragana key for Japanese USB word-processing keyboards.
32) Usage 0x94 (Keyboard LANG5) "Defines the Zenkaku/Hankaku key for Japanese USB word-processing keyboards.
33) The symbol displayed will depend on the current locale settings of the operating system. For example, the US thousands separator would be a comma, and the decimal separator would be a period.

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
34) The symbol displayed will depend on the current locale settings of the operating system. For example the US currency unit would be $ and the sub-unit would be ¢.							

Appendix G Modifier Byte Definitions

This appendix is from the following document found on www.usb.org: ***Device Class Definition for Human Interface Devices (HID) Version 1.11***, and specifically for this manual, ***Section 8.3 Report Format for Array Items***.

The modifier byte is defined as follows:

Table 6-4 - Modifier Byte Definitions

Bit	Key
0	LEFT CTRL
1	LEFT SHIFT
2	LEFT ALT
3	LEFT GUI
4	RIGHT CTRL
5	RIGHT SHIFT
6	RIGHT ALT
7	RIGHT GUI