USB HID Swipe Reader
PROGRAMMING REFERENCE MANUAL

PART NUMBER 99875378-2

MAY 2011

MAGTEK®

REGISTERED TO ISO 9001:2008

1710 Apollo Court
Seal Beach, CA 90740
Phone: (562) 546-6400
FAX: (562) 546-6301
Technical Support: (651) 415-6800
www.magtek.com

REVISIONS

Rev Number	Date	Notes
1.01	16 Jan 08	Initial Release
2.01	3 May 11	Added Track2Hash and TK2HashHex String

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 ABOVE ADDRESS 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 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 to be free from defects in material or workmanship under normal use. THE SOFTWARE IS PROVIDED AS IS WITHOUT 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 insure 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 ABOVE ADDRESS OR E-MAILED TO support@magtek.com.

TABLE OF CONTENTS

SECTION 1. FEATURES AND SPECIFICATIONS

INTRODUCTION

This document describes the properties, methods and events provided by the ActiveX component to communicate with MagTek Readers. For Additional Information, Please refer to the reader specific manual.

SECTION 2. PROPERTIES

NAME PROPERTY

Sets and returns the control name.

Syntax

USBHID.**Name** [= *value*]

The **Name** property syntax has these parts:

Part	Description
Value	A string value specifying the name of the control.

PORTOPEN PROPERTY

Sets and returns the status of the port.

Syntax

USBHID.**PortOpen** [= *value*]

The **PortOpen** property syntax has these parts:

Part	Description
Value	A Boolean value indicating the status of the port.

CARDDATA PROPERTY

Sets or returns the Card Data buffer for the USBHID control.

Syntax

USBHID.**CardData** [= *value*]

The **CardData** property syntax has these parts:

Part	Description
Value	A String value indicating the card data.

CARDDATAMASKED PROPERTY

Sets or returns the Masked Card Data buffer if supported by the reader for the USBHID control.

Syntax

USBHID.**CardDataMasked** [= *value*]

The **CardDataMasked** property syntax has these parts:

Part	Description
Value	A String value indicating the masked card data.

TRACK1LENGTH PROPERTY

Sets or returns the length of track1 data for the USBHID control.

Syntax

USBHID.Track1Length [= *value*]

The **Track1Length** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of track1 data.

TRACK2LENGTH PROPERTY

Sets or returns the length of track2 data for the USBHID control.

Syntax

USBHID.Track2Length [= *value*]

The **Track2Length** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of track2 data.

TRACK3LENGTH PROPERTY

Sets or returns the length of track3 data for the USBHID control.

Syntax

USBHID.Track3Length [= *value*]

The **Track3Length** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of track3 data.

TRACK1LENGTHMASKED PROPERTY

Sets or returns the length of masked track1 data if supported by the reader for the USBHID control.

Syntax

USBHID.Track1LengthMasked [= *value*]

The **Track1LengthMasked** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of masked track1 data.

TRACK2LENGTHMASKED PROPERTY

Sets or returns the length of masked track2 data if supported by the reader for the USBHID control.

Syntax

USBHID.Track2LengthMasked [= *value*]

The **Track2LengthMasked** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of masked track2 data.

TRACK3LENGTHMASKED PROPERTY

Sets or returns the length of masked track3 data if supported by the reader for the USBHID control.

Syntax

USBHID.Track3LengthMasked [= *value*]

The **Track3LengthMasked** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of masked track3 data.

MPRINTLEN PROPERTY

Sets or returns the length of MagnePrint if supported by the reader for the USBHID control.

Syntax

USBHID.MPrintLen [= *value*]

The **MPrintLen** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the length of MagnePrint data.

MPRINTSTATUS PROPERTY

Sets or returns the status of MagnePrint data if supported by the reader for the USBHID control.

Syntax

USBHID.MPrintStatus [= *value*]

The **MPrintStatus** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the status of MagnePrint data.

MPRINTDATA PROPERTY

Sets or returns the MagnePrint data if supported by the reader for the USBHID control.

Syntax

USBHID.MPrintData [= *value*]

The **MPrintData** property syntax has these parts:

Part	Description
Value	A Variant value indicating the content of MagnePrint data.

SEQUENCENUMBER PROPERTY

Sets or returns the sequence number if supported by the reader for the USBHID control.

Syntax

USBHID. SequenceNumber [= *value*]

The **SequenceNumber** property syntax has these parts:

Part	Description
Value	A Numeric value indicating the sequence number from the reader.

SESSIONID PROPERTY

Sets or returns the session id if supported by the reader for the USBHID control.

Syntax

USBHID. SessionID[= *value*]

The **SessionID** property syntax has these parts:

Part	Description
Value	A Variant value indicating the session id from the reader.

TRACK2HASH PROPERTY

Sets or returns the track 2 hash value if supported by the reader for the USBHID control.

Syntax

USBHID. Track2Hash[= *value*]

The **Track2Hash** property syntax has these parts:

Part	Description
Value	A Variant value indicating the Track 2 Hash value from the reader.

DUKPTKSN PROPERTY

Sets or returns the key serial number if supported by the reader for the USBHID control.

Syntax

USBHID. DUKPTKSN[= *value*]

The **DUKPTKSN** property syntax has these parts:

Part	Description
Value	A String value indicating the key serial number from the reader.

DEVICESERIALNUM PROPERTY

Sets or returns the device serial number if supported by the reader for the USBHID control.

Syntax

USBHID. DeviceSerialNum[= *value*]

The **DeviceSerialNum** property syntax has these parts:

Part	Description
Value	A String value indicating the device serial number from the reader.

CARDENCODETYPE PROPERTY

Sets or returns the card encode type if supported by the reader for the USBHID control.

Syntax

USBHID. CardEncodeType[= *value*]

The **CardEncodeType** property syntax has these parts:

Part	Description
Value	A String value indicating the card encode type from the reader.

Remarks

Explanation of tracks:

Values	Description
ISO	ISO Card Type
AAMVA	AAMVA Card Type
CADV	California Driver License
BANK	Bank Card Type
OTHER	Other Card Type
UNDETERMINED	Undetermined Card Type
NONE	No Card Type Detected

SWIPECOMMANDOUTPUT PROPERTY

Saves the output of the last function call to USBSwipe_Command or USBSwipe_CommandWithLength.

Syntax

USBHID. SwipeCommandOutput [= *value*]

The **SwipeCommandOutput**property syntax has these parts:

Part	Description
Value	A String value indicating the output of the last swipe command sent to the reader.

READERID PROPERTY

Sets or returns the ReaderID for the USBHID control.

Syntax

USBHID. ReaderID [= *value*]

The **ReaderID** property syntax has these parts:

Part	Description
Value	A String value indicating the Reader Product ID.

SECTION 3. METHODS

GETTRACK METHOD

Returns card data from a specific track.

Syntax

Result =*USBHID*.**GetTrack**(*TrackNum As Integer*) As String

The **GetTrack** method syntax has these parts:

Part	Description
TrackNum	An integer specifying which Track Number from Card Data to Return.

Remarks

Explanation of tracks:

Track #	Description
0	Unparsed Card Data. Same as '.CardData'. It includes all tracks.
1	Start Sentinel '%' End Sentinel is the first '?' after Start Sentinel.
2	Start Sentinel ';' End Sentinel is the first '?' after Start Sentinel.
3	Start Sentinel '+' End Sentinel is the first '?' after Start Sentinel.
4	Start Sentinel '!' End Sentinel is the first '?' after Start Sentinel (used in AAMVA cards).
5	Start Sentinel '#' End Sentinel is the first '?' after Start Sentinel (used in CADL's).

Data Type

String

GETTRACKMASKED METHOD

Returns masked card data if supported by reader from a specific track.

Syntax

Result =*USBHID*.**GetTrackMasked**(*TrackNum As Integer*) As String

The **GetTrackMasked** method syntax has these parts:

Part	Description
TrackNum	An integer specifying which Track Number from masked card data to Return.

Remarks

Explanation of tracks:

Track #	Description
0	Unparsed Card Data. Same as '.CardData'. It includes all tracks.
1	Start Sentinel '%' End Sentinel is the first '?' after Start Sentinel.
2	Start Sentinel ';' End Sentinel is the first '?' after Start Sentinel.
3	Start Sentinel '+' End Sentinel is the first '?' after Start Sentinel.
4	Start Sentinel '!' End Sentinel is the first '?' after Start Sentinel (used in AAMVA cards).
5	Start Sentinel '#' End Sentinel is the first '?' after Start Sentinel (used in CADL's).

Data Type

String

GETFNAME METHOD

Returns the First Name from the track data.

Syntax

Result =*USBHID*.**GetFName**()

Remarks

Returns the First Name on an ABA type ‘A’ or ‘B’ formatted card or returns nothing.

Data Type

String

GETFNAMEMASKED METHOD

Returns the First Name from the masked track data if supported by the reader.

Syntax

Result =*USBHID*.**GetFNameMasked**()

Remarks

Returns the First Name on an ABA type ‘A’ or ‘B’ formatted card or returns nothing.

Data Type

String

GETLNAME METHOD

Returns the Last Name from the track data.

Syntax

Result =*USBHID*.**GetLName**()

Remarks

Returns the Last Name on an ABA type ‘A’ or ‘B’ formatted card or returns nothing.

Data Type

String

GETLNAMEMASKED METHOD

Returns the Last Name from the masked track data if supported by the reader.

Syntax

Result =*USBHID*.**GetLNameMasked** ()

Remarks

Returns the Last Name on an ABA type ‘A’ or ‘B’ formatted card or returns nothing.

Data Type

String

MAGNEPRINTDATAHEXSTRING METHOD

Returns the Magneprint data if supported by the reader in 2 Byte hex format.

Syntax

Result =*USBHID*.**MagnePrintDataHexString**()

Remarks

Returns the MagnePrint data as a 2 byte hex.

Data Type

String

SESSIONIDHEXSTRING METHOD

Returns the session id data if supported by the reader in 2 Byte hex format.

Syntax

Result =*USBHID*.**SessionIDHexString**()

Remarks

Returns the Session ID data as a 2 byte hex.

Data Type

String

TK2HASHHEXSTRING METHOD

Returns the track 2 hash data if supported by the reader in 2 Byte hex format.

Syntax

Result =*USBHID*.**TK2HashHexString**()

Remarks

Returns the Track2Hash data as a 2 byte hex.

Data Type

String

USBSWIPE_COMMAND METHOD

Allows direct communication to the reader.

Syntax

Result =*Result = USBHID*.**USBSwipe_Command**(InPut,OutPut)

Remarks

Send specific commands to the reader. The length of InPut is automatically calculated by the component. The response is sent through Output and also gets saved on the property SwipeCommandOutput.

Data Type

String[In],Variant[Out],Long[Result]

USBSWIPE_COMMANDWITHLENGTH METHOD

Allows direct communication to the reader.

Syntax

Result =*Result = USBHID*.**USBSwipe_CommandWithLength**(InPut,OutPut)

Remarks

Send specific commands to the reader. The length of InPut is not automatically calculated by the component. It must be specified by the caller. The response is sent through Output and also gets saved on the property SwipeCommandOutput.

Data Type

String[In],Variant[Out],Long[Result]

FINDELEMENT METHOD

Returns particular information after parsing out specific card data, based on parameters selected by the user.

Syntax

Result =*USBHID*.FindElement (*ByVal TrackNum As Integer, ByVal RefChar As String, ByVal Displacement As Integer, ByVal NumDigits As String,* [*DirectionBack As Variant*]) As String

Part	Description
TrackNum	See Values From GetTrack method.
RefChar	The Character to Reference - Can also include multiples (e.g., ^^) would indicate Reference Character is the second ^.
Displacement	Number of Digits to Displace.
NumDigits	The Length or Terminating Character - Can also include multiples (e.g., = = =) would indicate Terminating Character is the Third =.
DirectionBack	The Direction to Travel. *This parameter is optional.*

Example

Assume CardData =
‘%B1234567890123456^CARD/TEST^9912101?;1234567890123456=9912101?

USBHID.FindElement(2,”;”,0,”=”) would return ‘1234567890123456’

USBHID.FindElement(2,”;”,0,”16”) would return ‘1234567890123456’

USBHID.FindElement(1,”^”,0,”^^”) would return ‘CARD/TEST’

USBHID.FindElement(1,”^”,0,”/”) would return ‘CARD’

DirectionBack: If omitted DirectionBack is False (Left to Right) when specifying a reference character of anything but an End Sentinel (?). If the Reference Character is an End Sentinel then DirectionBack is True (Right to Left). If specified, then the actual value takes precedence.

Data Type

String

FINDELEMENTMASKED METHOD

Returns particular information after parsing out specific masked card data, based on parameters selected by the user.

Syntax

Result =*USBHID*.FindElementMasked (*ByVal TrackNum As Integer*, *ByVal RefChar As String*, *ByVal Displacement As Integer*, *ByVal NumDigits As String*, [*DirectionBack As Variant*]) As String

Part	Description
TrackNum	See Values From GetTrack method.
RefChar	The Character to Reference - Can also include multiples (e.g., ^^) would indicate Reference Character is the second ^.
Displacement	Number of Digits to Displace.
NumDigits	The Length or Terminating Character - Can also include multiples (e.g., = = =) would indicate Terminating Character is the Third =.
DirectionBack	The Direction to Travel. *This parameter is optional.*

Example

Assume CardData =
‘%B1234567890123456^CARD/TEST^9912101?;1234567890123456=9912101?

USBHID.FindElementMasked(2,”;”,0,”=”) would return ‘1234567890123456’

USBHID.FindElementMasked(2,”;”,0,”16”) would return ‘1234567890123456’

USBHID.FindElementMasked(1,”^”,0,”^^”) would return ‘CARD/TEST’

USBHID.FindElementMasked(1,”^”,0,”/”) would return ‘CARD’

DirectionBack: If omitted DirectionBack is False (Left to Right) when specifying a reference character of anything but an End Sentinel (?). If the Reference Character is an End Sentinel then DirectionBack is True (Right to Left). If specified, then the actual value takes precedence.

Data Type

String

GETDEFSETTING METHOD

Returns data from the registry.

Syntax

Result =*USBHID*.**GetDefSetting**(*ByVal Key As String*, [*ByVal Default As String*]) As String

The **GetDefSetting** method syntax has the following parts:

Part	Description
Key	Registry Key
Default	Value returned if the key does not exist. *This parameter is optional.*

Remarks

This method returns data stored in the HKEY_CURRENT_USER/Software/VB and VBA Program Settings/USBHID/Startup section of the registry. If the *Key* value doesn't exist within the section, the *Default* value is returned.

Example

USBHID.**GetDefSetting**("CommPort", "2")

This returns the value of the CommPort Key. If the CommPort key doesn't exist, '2' is returned.

Data Type

String

SAVEDEFSETTING METHOD

Saves data to the registry.

Syntax

Result =*USBHID*.**SaveDefSetting**(*Key As String, Setting As String*)

The **SaveDefSetting** method syntax has the following parts:

Part	Description
Key	Registry Key
Setting	Value stored in the registry key.

Remarks

This method stores data in the HKEY_CURRENT_USER/Software/VB and VBA Program Settings/USBHID/Startup section of the registry.

Example

USBHID.**SaveDefSetting**(“CommPort”, “4”)

This stores the value of “4” in the CommPort Registry Key.

ABOUT METHOD

Displays a message box containing information specific to the USBHID.

Syntax

Result =*USBHID*.**About**

CLEARBUFFER METHOD

Syntax

Result =*USBHID*.**ClearBuffer**()

Remarks

This method sets the properties to their default value.

SECTION 4. EVENTS

CARDDATACHANGED EVENT

This event fires when a card with magnetic data is passed through the reader and the PortOpen property is True.

Syntax

Sub USBHID_**CardDataChanged()**

CARDDATAERROR EVENT

This event fires when an error in card swipe is detected and the PortOpen property is True.

Syntax

Sub USBHID_**CardDataError()**