

Secure Card Reader Authenticator API PROGRAMMING REFERENCE MANUAL

MANUAL PART NUMBER 99875535-1

FEBRUARY 2011

MAGTEK[®]

REGISTERED TO ISO 9001:2008

1710 Apollo Court

Seal Beach, CA 90740

Phone: (562) 546-6400

Technical Support: (651) 415-6800

www.magtek.com

Copyright© 2004 – 2011

MagTek® , Inc.

Printed in the United States of America

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

MagTek is a registered trademark of MagTek, Inc.

Microsoft® is a trademark of Microsoft, Inc.

REVISIONS

Rev Number	Date	Notes
1.01	15 Feb 2011	Initial Release

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 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 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 ABOVE ADDRESS, OR E-MAILED TO support@magtek.com.

TABLE OF CONTENTS

SECTION 1. OVERVIEW	1
SCRA METHODS	1
SCRA EVENTS	1
SCRA ENUMS	1
SECTION 2. SECURE CARD READER AUTHENTICATOR API.....	3
MTUSCRAOPENDEVICE	3
MTUSCRACLOSEDEVICE	3
MTUSCRASENDCOMMAND	3
MTUSCRAGETCARDDATA	4
MTUSCRAGETCARDDATASTR	5
MTUSCRACLEARBUFFER	5
MTUSCRACARDDATASTATECHANGEDNOTIFY	6
MTUSCRADEVICESTATECHANGEDNOTIFY	6
MTUSCRAGETDEVICESTATE	7
MTUSCRAGETCARDDATASTATE.....	7
MTUSCRAGETPID	7

SECTION 1. OVERVIEW

SCRA METHODS

MTUSCRAOpenDevice	Opens a SCRA Swipe Reader
MTUSCRACloseDevice	Closes opened SCRA Swipe Reader
MTUSCRASendCommand	Allows sending command to the SCRA Swipe directly
MTUSCRAGetCardData	Allows retrieving card data through structures
MTUSCRAGetCardDataStr	Allows retrieving card data as a buffer with field delimiters
MTUSCRAClearBuffer(void);	Clears existing card buffer
MTUSCRACardDataStateChangedNotify	Sets a callback function to notify card data state change
MTUSCRADeviceStateChangedNotify	Sets a callback function to notify device state change
MTUSCRAGetDeviceState	Retrieves the current device state
MTUSCRAGetCardDataState	Retrieves the current card data state
MTUSCRAGetPID	Retrieves the Product ID of the SCRA swipe reader

SCRA EVENTS

```
typedef void (WINAPI *CallBackCardDataStateChanged)(DWORD lpdwCardDataState);
CallBackCardDataStateChanged
```

Receives information of Card Data States. Possible values are defined under ECardDataStateValues enum

```
typedef void (WINAPI *CallBackDeviceStateChanged)(DWORD lpdwDeviceState);
CallBackDeviceStateChanged
```

Receives information on Device States. Possible values are defined under EDeviceStateValues enum

SCRA ENUMS

EErrorValues	MTSCRA_ST_OK=0, MTSCRA_ST_FAILED=1, MTSCRA_ST_OPEN=2, MTSCRA_ST_INVALID_PARAM=3
EDeviceStateValues	MTSCRA_STATE_DISCONNECTED=0, MTSCRA_STATE_CONNECTED=1, MTSCRA_STATE_ERROR=2
ECardReadValues	MTSCRA_CARDREAD_OK=0, MTSCRA_CARDREAD_ERROR=1
ECardDataStateValues	MTSCRA_DATA_NOTREADY=0, MTSCRA_DATA_READY=1, MTSCRA_DATA_ERROR=2

SECTION 2. SECURE CARD READER AUTHENTICATOR API

MTUSCRAOPENDEVICE

This function opens a SCRA Swipe Reader

Syntax

```
MTUSCRA_API DWORD WINAPI MTUSCRAOpenDevice(LPSTR lpDeviceName);
```

Parameter

lpDeviceName

Name of the device to open for future use. Current users should pass an empty string.

Return Values:

Please see EErrorValues

MTUSCRACLOSEDEVICE

This function closes currently opened SCRA Swipe Reader

Syntax

```
MTUSCRA_API DWORD WINAPI MTUSCRACloseDevice();
```

Return Values:

Please see EErrorValues

MTUSCRASENDCOMMAND

This function Allows sending command to the SCRA Swipe dreader

Syntax

```
MTUSCRA_API DWORD WINAPI MTUSCRASendCommand(LPSTR lpCmd, DWORD  
lpdwCmdLen,LPSTR lpResult, DWORD lpdwResultLen);
```

Parameter

lpCmd

Command to send to the device

lpResult

Buffer to receive result

lpdwCmdLen

Length of the Command

DWORD lpdwResultLen

Size of lpResult

Return Values:

Please see ErrorValues

MTUSCRAGETCARDATA

This function retrieves card data information through a predefined structure

Syntax

```
MTUSCRA_API DWORD WINAPI MTUSCRAGetCardData(MTMSRDATA* lpMTMSRDATA);
```

Parameter

MTMSRDATA

MSR Data Structure

Return Values:

Please see EErrorValues

```
typedef struct _MTMSRDATA
```

```
{
```

```
    char m_szCardData[DEF_MSR_DATA_LEN * 3]; //Card Data
    char m_szCardDataMasked[DEF_MSR_DATA_LEN * 3]; //Masked Card Data
    char m_szTrack1Data[DEF_MSR_DATA_LEN]; //Track 1 Data
    char m_szTrack2Data[DEF_MSR_DATA_LEN]; //Track 2 Data
    char m_szTrack3Data[DEF_MSR_DATA_LEN]; //Track 3 Data
    char m_szTrack1DataMasked[DEF_MSR_DATA_LEN]; //Masked Track 1 Data
    char m_szTrack2DataMasked[DEF_MSR_DATA_LEN]; //Masked Track 2 Data
    char m_szTrack3DataMasked[DEF_MSR_DATA_LEN]; //Masked Track 3 Data
    char m_szMagnePrintData[DEF_MSR_DATA_LEN]; //MagnePrint Data
    char m_szCardEncodeType[DEF_MSR_DATA_LEN]; //Card Encode Type
    char m_szMagnePrintStatus[DEF_MSR_DATA_LEN]; //MagnePrint Status as String
    char m_szDUKPTSessionID[DEF_MSR_DATA_LEN]; //DUKPT Session ID
    char m_szDeviceSerialNumber[DEF_MSR_DATA_LEN]; //Device Serial Number
    char m_szDUKPTKSN[DEF_MSR_DATA_LEN]; //DUKPT Key Serial Number
    char m_szFirstName[DEF_MSR_DATA_LEN]; //First Name From Track 1
    char m_szLastName[DEF_MSR_DATA_LEN]; //Last Name From Track 1
    char m_szPAN[DEF_MSR_DATA_LEN]; //PAN From Track 2
    char m_szMonth[DEF_MSR_DATA_LEN]; //Expiration Month
    char m_szYear[DEF_MSR_DATA_LEN]; //Expiration Year

    DWORD m_dwReaderID; //Reader Product ID
    DWORD m_dwMagnePrintLength; //MagnePrint Length
    DWORD m_dwMagnePrintStatus; //MagnePrint Status
    DWORD m_dwTrack1Length; //Track 1 Data Length
    DWORD m_dwTrack2Length; //Track 2 Data Length
    DWORD m_dwTrack3Length; //Track 3 Data Length
    DWORD m_dwTrack1LengthMasked; //Masked Track 1 Data Length
    DWORD m_dwTrack2LengthMasked; //Masked Track 2 Data Length
    DWORD m_dwTrack3LengthMasked; //Masked Track 3 Data Length
    DWORD m_dwCardEncodeType; //Card Encode Type
    DWORD m_dwTrack1DcdStatus; //Track 1 Decode Status
    DWORD m_dwTrack2DcdStatus; //Track 2 Decode Status
    DWORD m_dwTrack3DcdStatus; //Track 3 Decode Status
    DWORD m_dwCardSwipeStatus; //Card Swipe Status, ECardReadValues.
```

```
} MTMSRDATA, *PMTMSRDATA;
```


MTUSCRAGETCARDATASTR

This function retrieves card data information through a predefined string and field separator

Syntax

```
MTUSCRA_API DWORD WINAPI MTUSCRAGetCardDataStr(LPSTR lpStrData,LPSTR  
lpStrFieldDelimiter);
```

Parameter

lpStrData

Buffer to receive Data

lpStrFieldDelimiter

Delimiter to separate the data fields

Return Values:

Please see EErrorValues

Fields:

Device ID, Device Serial Number, Card Swipe Status, CardEncode Type, Track 1 Decode Status, Track 2 Decode Status, Track 3 Decode Status, MagnePrint Status, Track 1 Length, Track 2 Length, Track 3 Length, Masked Track 1 Length, Masked Track 2 Length, Masked Track 3 Length, MagnePrint Length, Card Data, Masked Card Data, DUKPT Session ID, DUKPT Key Serial Number, First Name, Last Name, PAN, Month, Year, Track 1 Data, Track 2 Data, Track 3 Data, Masked Track 1 Data, Masked Track 2 Data, Masked Track 3 Data, MagnePrint Data,

MTUSCRACLEARBUFFER

This function clears the card data buffer

Syntax

```
MTUSCRA_API void WINAPI MTUSCRAClearBuffer(void);
```

Return Values:

None

MTUSCRACARDDATASTATECHANGEDNOTIFY

This function sets a callback function to notify card data states.

Syntax

TUSCRA_API void WINAPI

MTUSCRACardDataStateChangedNotify(CallBackCardDataStateChanged lpFuncNotify);

Parameter

lpFuncNotify

Function to call to provide notification information

Return Values:

None

MTUSCRADEVICESTATECHANGEDNOTIFY

This function sets a callback function to notify device states. Please refer to enum section for possible values.

Syntax

MTUSCRA_API void WINAPI MTUSCRADeviceStateChangedNotify(CallBackDeviceStateChanged lpFuncNotify);

Parameter

lpFuncNotify

Function to call to provide notification information

Return Values:

None

MTUSCRAGETDEVICESTATE

This function retrieves current device state information. For details on device states, please refer to the EDeviceStateValues

Syntax

```
MTUSCRA_API void WINAPI MTUSCRAGetDeviceState(DWORD* lpdwDeviceState) ;
```

Return Values:

None

MTUSCRAGETCARDDATASTATE

This function retrieves current device state information. For details on device states, please refer to the ECardDataStateValues

Syntax

```
MTUSCRA_API void WINAPI MTUSCRAGetCardDataState(DWORD* lpdwCardDataState) ;
```

Return Values:

None

MTUSCRAGETPID

This function retrieves the current product id of the SCRA Swipe Reader

Syntax

```
MTUSCRA_API void WINAPI MTUSCRAGetPID(DWORD* lpdwPID) ;
```

Return Values:

None

