

USB READER PROGRAMMING REFERENCE MANUAL FOR ANDROID

PART NUMBER 99875573-1

September 2011

Confidential

This document contains the proprietary information of MagTek. Its receipt or possession does not convey any rights to reproduce or disclose its contents or to manufacture, use or sell anything it may describe. Reproduction, disclosure or use without specific written authorization of MagTek is strictly forbidden.

Unpublished – All Rights Reserved

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

Copyright© 2001-2011
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.

REVISIONS

Rev Number	Date	Notes
1.01	16 Sep 11	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. libDynamag Class	1
Methods.....	2
openDevice:	2
closeDevice:.....	2
isDeviceConnected	2
sendCommand.....	3
setCardData:.....	3
clearCardData	3
getCardData	3
getMaskedTracks	4
getTrack1Masked.....	4
getTrack2Masked.....	4
getTrack3Masked.....	4
getTrack1	5
getTrack2	5
getTrack3	5
getMagnePrint.....	5
getMagnePrintStatus	5
getDeviceSerial	6
getSessionID	6
getKSN.....	6
sendCommandWithLength	6
Section 2. Code Examples	7
Open Device:	7
Close Device:	7
Handler for Processing Messages From Reader:	8

SECTION 1. libDynamag CLASS

Classes	Description
libDynamag	This class allows you to perform reader functions.

Methods:

openDevice	Open device
closeDevice	Close device
isDeviceConnected	Check the connection status of reader
sendCommand	Send command to reader
setCardData	Set card data that is retrieved from reader
clearCardData	Clear card data that is stored through the setCardData function
getCardData	Retrieves the existing stored card data
getMaskedTracks	Retrieves the existing stored masked track data
getTrack1	Retrieves encrypted track1
getTrack2	Retrieves encrypted track2
getTrack3	Retrieves encrypted track3
getTrack1Masked	Retrieves masked track1
getTrack2Masked	Retrieves masked track2
getTrack3Masked	Retrieves masked track3
getMagnePrint	Retrieves encrypted MagnePrint
getMagnePrintStatus	Retrieves encrypted MagnePrintStatus
getDeviceSerial	Retrieves device serial number
getSessionID	Retrieves session ID
getKSN	Retrieves key serial number

Methods

openDevice:

This function open the reader.

```
- public void openDevice()
```

Parameters

None

Return Value

None

closeDevice:

This function close the reader.

```
- public void closeDevice(){
```

Parameters

None

Return Value

None

isDeviceConnected

This function retrieves the connection status of the reader.

```
- public bool isDeviceConnected()
```

Parameters

None

Return Value

TRUE if the device is connected. Otherwise, return FALSE.

sendCommand

This function send command to reader.

```
- public byte[] sendCommand(CharSequence cmd)
```

Parameters

Cmd

A string command sent to reader. For example, command “0003” (Where “00” is command number, “03” is property ID) will retrieve device serial number from reader.

Return Value

A response returned from reader.

For more details on other commands, please refer to COMMANDS section in “MagneSafe Communication reference manual”.

setCardData:

Set card data that is retrieved from reader through getTracksData method for parsing.

```
- public void setCardData(string lpCardData)
```

Parameters

pCardData

A track data string.

Return Value

clearCardData

Clear card data that is set through setCardData method.

```
- public void clearCardData()
```

Parameters

Return Value

getCardData

Get existing stored card data.

```
- public string getCardData()
```

Parameters

Return Value

Return stored card data string.

getMaskedTracks

Get stored masked tracks data.

- `public string getMaskedTracks()`

Parameters

Return Value

Return stored masked tracks data string.

getTrack1Masked

Get stored masked track1 data.

- `public string getTrack1Masked()`

Parameters

Return Value

Return stored masked track1 data string.

getTrack2Masked

Get stored masked track2 data.

- `public string getTrack2Masked()`

Parameters

Return Value

Return stored masked track2 data string.

getTrack3Masked

Get stored masked track3 data.

- `public string getTrack3Masked()`

Parameters

Return Value

Return stored masked track3 data string.

getTrack1

Get stored encrypted track1 data.

- `public string getTrack1()`

Parameters

Return Value

Return stored masked track1 data string.

getTrack2

Get stored encrypted track2 data.

- `public string getTrack2()`

Parameters

Return Value

Return stored masked track2 data string.

getTrack3

Get stored encrypted track3 data.

- `public string getTrack3()`

Parameters

Return Value

Return stored masked track3 data string.

getMagnePrint

Get stored MagnePrint.

- `public string getMagnePrint()`

Parameters

Return Value

Return stored MagnePrint string.

getMagnePrintStatus

Get stored MagnePrintStatus.

- `public string getMagnePrintStatus()`

Parameters

Return Value

Return stored MagnePrintStatus string.

getDeviceSerial

Get stored device serial number.

```
- public string getDeviceSerial()
```

Parameters

Return Value

Return stored device serial number.

getSessionID

Get stored session ID.

```
- public string getSessionID()
```

Parameters

Return Value

Return stored session ID.

getKSN

Get stored key serial number.

```
- public string getKSN()
```

Parameters

Return Value

Return stored key serial number.

sendCommandWithLength

This function send command to reader.

```
- public byte[] sendCommandWithLength(CharSequence cmd)
```

Parameters

Cmd

A string command sent to reader. For example, command "000103" (Where "00" is command number, "01" is the length and "03" is property ID) will retrieve device serial number from reader.

Return Value

A response returned from reader.

For more details on other commands, please refer to COMMANDS section in "MagneSafe Communication reference manual"

SECTION 2. CODE EXAMPLES

Open Device:

```
@Override
public synchronized void onResume() {
    super.onResume();

    if (! MagTeklibQwickey.isConnected()) {
        MagTeklibQwickey.openDevice();
    }
}
```

Close Device:

```
@Override
public void onStop() {
    super.onStop();

    if (MagTeklibQwickey.isConnected()) {
        MagTeklibQwickey.closeDevice();
    }
}
```

Handler for Processing Messages From Reader:

```
private class MtHandlerCallback implements Callback {
    public boolean handleMessage(Message msg) {

        boolean ret = false;

        switch (msg.what) {
            case DEVICE_MESSAGE_CARDDATA_CHANGE:
                mStringCardDataBuffer = (String)msg.obj;
                // Process card data
                ret = true;
                break;

            case DEVICE_STATUS_CONNECTED:
                if (((Number)msg.obj).intValue() ==
                    DEVICE_STATUS_CONNECTED_SUCCESS) {

                    // Reader connected successfully

                } else if (((Number)msg.obj).intValue() == D
                    EVICE_STATUS_CONNECTED_FAIL){

                    // Reader connection failed

                } else if (((Number)msg.obj).intValue() ==
                    DEVICE_STATUS_CONNECTED_PERMISSION_DENIED){

                    // Reader connection permission denied

                }

                break;

            case DEVICE_STATUS_DISCONNECTED:

                // Reader disconnected

                break;

            default:
                ret = false;
                break;

        }

        return ret;
    }
}
```