JMSR Applet for
MagTek HID Card Reader
PROGRAMMER'S GUIDE

PART NUMBER 99875539 -1

MARCH 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	25 Mar 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

I. INTRODUCTION

JMSR Applet transfers Card data from MagTek USB Card Reader device to a host program. The applet is a signed/trusted applet.

JMSR Applet supports device with PID 0x0011, 0x0013, 0x0002, and 0x0003.

II. SYSTEM REQUIREMENTS

The following are general requirements for running JMSR applet on the following supported platforms:

Windows OS: Windows XP SP2 and Vista
Java: JVM 1.6 (32-bit) and later.
Browsers: Internet Explorer 7, Firefox version 3.0 and later.

There is no specific memory or hardware requirements.

III. SOFTWARE REQUIREMENTS

JMSR applet was tested with Mozilla Firefox browser version 3.0.1 and Internet Explorer browser version 7 running on Windows XP platform with Java version 1.6.

JMSR applet requires library file MTHIDMCR.DLL.

If MTHIDMCR.DLL is not present in the system, the applet automatically downloads the file onto Windows System directory.

IV. JMSR COMPONENT REQUIREMENTS

The directory on the host Web Server must contain the following four components and these four files must be in the same directory.

1. Jar file for JMSR Applet

JMTCardReader.jar

2. Windows DLL file

MTHIDMCR.DLL

3. HTML Page

In order to use JMSR Applet, the applet is embedded in a HTML page inside <Applet> tag for Mozilla Firefox browser or <Object> tag for Internet Browser. For example:

```
<applet
type="application/x-java-applet;version=1.6"
                        codebase=”http://www.testserver.com”
                        code="JMTCardReader.class"
                        name="JMSR"
                        archive="JMTCardReader.jar"
                        scriptable = "true"
                        style = "visibility:hidden;"
                        mayscript="mayscript"
                        pluginspage = "http://java.com/en/download/index.jsp"
>
                        <param name="cache_option" value="No">
                        <param name="classloader_cache" value="true">

 </applet>
```

The codebase parameter must be set to the URL where the applet is located. In the example, the codebase is set to http://www.testserver.com where the server “testserver.com“ contains all four components described in this section.

The classloader_cache parameter must be set to true to avoid Java IO exception when JMSR applet is destroyed and restarted in the same browser’s session such as when a browser is refresh.

4. JavaScript functions

When a card is swiped through Card Reader device, the JMSR Applet sends card data to the browser by calling several JavaScript functions. The following JavaScript functions will be called by the JMSR applet every time new data is collected from Card Reader device:

SetCardData(*value*)

When this function is called by JMSR applet, it receives the card data.

SetCardDataMsk(*value*)

When this function is called by JMSR applet, it receives the masked card data.

SetPAN(*value)*

When this function is called by JMSR applet, it receives the PAN number.

SetFirstName(*value*)

When this function is called by JMSR applet, it receives the First Name from the track data.

SetLastName(*value*)

When this function is called by JMSR applet, it receives the Last Name from the track data.

SetMonth(*value*)

When this function is called by JMSR applet, it receives the month from the track data.

SetYear(*value*)

When this function is called by JMSR applet, it receives the year from the track data.

SetTrack1Len(*value*)

When this function is called by JMSR applet, it receives the length of track1 data.

SetTrack2Len(*value*)

When this function is called by JMSR applet, it receives the length of track2 data.

SetTrack3Len(*value*)

When this function is called by JMSR applet, it receives the length of track3 data.

SetTrack1MskLen(*value*)

When this function is called by JMSR applet, it receives the length of masked track1 data.

SetTrack2MskLen(*value*)

When this function is called by JMSR applet, it receives the length of masked track2 data.

SetTrack3MskLen(*value*)

When this function is called by JMSR applet, it receives the length of masked track3 data.

SetTrk1Encrypted(*value*)

When this function is called by JMSR applet, it receives track 1 encrypted data.

SetTrk2Encrypted(*value*)

When this function is called by JMSR applet, it receives track 2 encrypted data.

SetTrk3Encrypted(*value*)

When this function is called by JMSR applet, it receives track 3 encrypted.

SetTrack1MskData(*value*)

When this function is called by JMSR applet, it receives track 1 data masked.

SetTrack2MskData(*value*)

When this function is called by JMSR applet, it receives track 2 data masked.

SetTrack3MskData(*value*)

When this function is called by JMSR applet, it receives track 3 data masked.

SetMPStatus(*value*)

When this function is called by JMSR applet, it receives status of MagnePrint data.

SetMPData(*value*)

When this function is called by JMSR applet, it receives MagnePrint data

SetMPLen(*value*)

When this function is called by JMSR applet, it receives length of MagnePrint data.

SetDeviceSN(*value*)

When this function is called by JMSR applet, it receives serial number of the reader.

SetReaderID(*value*)

When this function is called by JMSR applet, it receives ID of the reader.

SetEncodeType(*value*)

When this function is called by JMSR applet, it receives the type of encoding.

SetDUKPTKSN(*value*)

When this function is called by JMSR applet, it receives the value of DUKPTKSN.

SetSessionID (*value*)

When this function is called by JMSR applet, it receives ID of the session.

SetCRStatus(*value*)

When this function is called by JMSR applet, it receives status of Card Reader. If Card Reader is connected, the status is "*Reader Connected*", otherwise, the status is "*Reader Disconnected*".

SetResults(*value*)

JMSR provides function SendStrCmd() to send a command value to Card Reader. JMSR applet uses this SetResults() to send to host the results JMSR receives from Card Reader.

SetAllCardData(*value*)

When this function is called by JMSR applet, it receives all Card Data in one single string. Individual data is separated by the character "|" .

Card data are arranged in the following order:

1. PAN
2. First Name
3. Last Name
4. Month
5. Year
6. Track 1 Length

7. Track 2 Length
8. Track 3 Length
9. Track 1 Masked Length
10. Track 2 Masked Length
11. Track 3 Masked Length
12. Track 1 Data
13. Track 2 Data
14. Track 3 Data
15. Track 1 Masked Data
16. Track 2 Masked Data
17. Track 3 Masked Data
18. MagnePrint Status
19. MagnePrint Data
20. MagnePrint Length
21. Device Serial Number
22. Card Reader ID
23. Encode Type
24. DUKPTKSN data
25. Session ID

SetCardReadingError()

If the card is swiped too fast or too slow or the card is damaged, card reading will fail. This function is called by JMSR applet to report card reading problem .

ReportJavaPluginVersion()

Before Java applet loading completes, the version of Java Virtual Machine is checked to ensure version 1.5 or later is installed.

DeviceReady()

This function is called when device is ready for card swiping. This function can be used as Device Ready event.

GetTrack1DecodeStatus()

This function returns the value of decode status for Track 1.

GetTrack2DecodeStatus()

This function returns the value of decode status for Track 2.

GetTrack3DecodeStatus()

This function returns the value of decode status for Track 3.

Methods

SendStrCmd()

Returns a string value from the reader.

Syntax
String SendCmdStr(*String strCmd*)

Remarks
Send specific command to the MTHID reader. The length of the *strCmd* is automatically calculated by the component. If the command fails or encounters an error, the error will be returned and the user needs to parse it.

The **SendData** method has this part:

Part	**Data Type**	**Description**
strCmd	String	Command

Example
String response = SendCmdStr (“0010”);
The response will be a hexadecimal string returned from the reader. An empty string returned indicates that the communication between the applet and the reader might most likely be disconnected.

SendCmdStrWithLength METHOD

Returns a string value from the reader.

Syntax
String SendCmdStrWithLength (*String strCmd*)

Remarks
Send specific command to the MTHID reader. The length of the *strCmd* is not automatically calculated by the component. If the command fails or encounters an error, the error will be return and the user needs to parse it.

The **SendCmdStrWithLength** method has this part:

Part	**Data Type**	**Description**
strCmd	String	Command

Example
String response = SendCmdStrWithLength (“000110”);

The response will be a hexadecimal string returned from the reader. An empty string returned indicates that the communication between the applet and the reader might most likely be disconnected.