

IPAD KEY INJECTION REFERENCE MANUAL

PART NUMBER 99875416-2

AUGUST 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Unpublished – All Rights Reserved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

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.
IPAD™ is a trademark of MagTek, Inc.

REVISIONS

Rev Number	Date	Notes
1.01	23 Jun 09	Initial Release
2.01	31 Aug 11	Modified cover page to be consistent with other IPAD docs; added higher-level steps documentation; modified commands 0x0B and 0x58 (Tables E and F); added documentation for MS2.0 formatting & fixed PIN key

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

IPAD KEY INJECTION PROCESS.....	1
OVERVIEW	1
GENERATE KEY PAIR, CREATE CSR, SEND TO MAGTEK	1
Generate Key Pair.....	1
Create CSR	1
Send to Magtek	1
BIND CERTIFICATE TO KEYLOADER	1
INJECT PIN AND MSR KEYS INTO THE IPAD.....	1
Send CRL to the IPAD.....	1
Get Device Certificate from the IPAD	1
Get Challenge from the IPAD	1
Derive DUKPT Key (or Use Fixed PIN Key)	2
Wrap Key with Device Public Key	2
Assemble Message and Sign with Keyloader Private Key	2
Send Message to IPAD.....	2
Get KCV, Compare with Calculated KCV.....	2
IPAD USB COMMUNICATIONS	3
HID USAGES	3
REPORT DESCRIPTOR	4
FEATURE REPORTS	7
Report 0x01 – Response ACK	7
Report 0x02 – End Session	8
Report 0x08 – Request Device Status	8
Report 0x09 – Set Device Configuration	9
Report 0x09 – Get Device Configuration.....	10
Report 0x0B – Get Challenge	10
Report 0x0C – Set Bitmap.....	11
Report 0x0E – Get Information.....	12
Report 0x10 – Send Big Block Data to Device.....	14
Report 0x58 – Request Device Cert.....	15
Report 0x58 – Key Handling or Manufacturing Command.....	15
INPUT REPORTS	19
Report 0x20 – Device State Report	19
Report 0x29 – Send Big Block Data to Host	19
APPENDIX A: STATUS AND MESSAGE TABLE	21

IPAD KEY INJECTION PROCESS

OVERVIEW

The keyloading host process shall generate a key pair and CSR, send the CSR to MagTek, and receive back a signed certificate. The keyloader shall then bind the certificate to the keyloader and inject the key into the IPAD.

GENERATE KEY PAIR, CREATE CSR, SEND TO MAGTEK

A unique certificate is required for each IPAD being keyed:

1. The keyloader host shall generate a key pair and its corresponding CSR
2. The keyloader host shall forward the CSR to the keyloader HSM
3. The keyloader host shall take the CSR from the HSM and send it to MagTek over a trusted channel
4. MagTek shall create a certificate from the CSR and send it back to the keyloader HSM.
5. The keyloader host shall take the certificate from the HSM and load it into the IPAD via the bind operation (see below)

Generate Key Pair

The keyloading host process shall generate a 2048 bit RSA key pair.

Create CSR

The keyloading host process shall create a PKCS #10 V1.7 CSR in DER format.

Send to Magtek

Process TBD.

BIND CERTIFICATE TO KEYLOADER

Before binding, the keyloading host process must send the latest CRL from MagTek to the IPAD. Please refer to the section “Send CRL to the IPAD” below.

The keyloading host process shall then send the certificate to the IPAD using feature report 0x10, and shall bind it using feature report 0x58 (byte 2 = 0x05 for the PIN cert).

INJECT PIN AND MSR KEYS INTO THE IPAD

Send CRL to the IPAD

The keyloading host process shall get the latest CRL from MagTek. (Process TBD.) The keyloading host process shall then send the CRL to the IPAD using feature report 0x10, and shall load it using feature report 0x58 (byte 2 = 0x0F for the PIN CRL). MagTek will send X.509 CRLs in DER format.

Get Device Certificate from the IPAD

The keyloading host process shall request the IPAD to send its device certificate to the host using feature report 0x58, with byte 1 = 0x02 and byte 2 = 0x01. The host shall then get the device cert using input report 0x29. The IPAD will send its device cert as an X.509 v3 certificate in DER format.

Get Challenge from the IPAD

The keyloading host process shall request the IPAD to send challenge information to the host using feature report 0x0B (byte 1 = 0x00 for the PIN key). The IPAD will return a feature report containing the device serial number in bytes 2-9 and a random token in bytes 10-13.

Derive DUKPT Key (or Use Fixed PIN Key)

The keyloading host process shall derive a DUKPT key as described in ANSI/X9 X9.24-1 Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques.

Alternately, you may use a static PIN key, which is an 8/16/24-byte DES key including a parity bit on each byte.

Wrap Key with Device Public Key

The keyloading host process shall wrap the derived DUKPT key (or the Fixed PIN key, if that option is used) with the Device Public Key using the PKCS#1 v2.1 RSAES-OAEP key transport algorithm (RFC 3447).

Assemble Message and Sign with Keyloader Private Key

The keyloading host process shall assemble a message containing the DUKPT key to be injected; it will then use feature report 0x10 to send the assembled message to the IPAD. Because the length of the assembled message exceeds 400 bytes and the feature report is limited to 60-byte data blocks, the host must parse the assembled message into multiple feature reports as required.

The keyloading host process shall sign the message with the Keyloader Private Key (a PKCS#1 v1.5 RSASSA-PKCS1-v1_5 signature using SHA-1).

Below is a graphic illustrating the format of the assembled message (before parsing) which corresponds to Table C (Table references may be found under the documentation of feature report 0x58; the examples given are for injecting the PIN DUKPT key).

Signature Length	Message Length	Header (Table D)				
Low,High (e.g. 0x00, 0x01)	Low,High (0x08, 0x00)	Header Length (0x15)	Keyid (Table E) (e.g. 0x00)	Device Serial Number	Random Token	Extra Field (Table F) (e.g. KSN)
2 bytes	2 bytes	1 byte	1 byte	8 bytes	4 bytes	8 bytes

Message (Table G)	Signature (Table H)
e.g. RSAES-OAEP key wrapped with device public key	Keyloader Private Key (see text above)
128 bytes	256 bytes

Send Message to IPAD

The keyloading host process shall send the assembled key message to the IPAD using command 0x10, and shall inject the key using command 0x58, with byte 2 = 0x15.

Get KCV, Compare with Calculated KCV

The keyloading host process shall request the IPAD to send the Key Check Value (KCV) to the host using feature report 0x0E, with byte 1 = 0x80.

IPAD USB COMMUNICATIONS

This device conforms to the USB specification revision 2.0 (compatible with 1.1). This device also conforms to the Human Interface Device (HID) class specification version 1.1. The IPAD communicates with the host as a vendor-defined HID device. The details about how the data and commands are structured into HID reports follow later in this document. The latest versions of the Windows operating systems come with a standard Windows USB HID driver.

Windows applications that communicate with this device can be easily developed using compilers such as Microsoft's Visual Basic or Visual C++. Such applications can interact with the device through API calls using the standard Windows USB HID driver, a basic component of all modern versions of the Windows operating system. A demonstration program that communicates with this device is available. This demo program can be used to test the device and it can be used as a guide for developing other applications. More details about the demo program follow later in this document.

It is recommended that application software developers become familiar with USB HID class specifications before attempting to communicate with this device. This document assumes that the reader is familiar with these specifications, which can be downloaded free at www.usb.org.

This is a full speed USB device. This device has some programmable configuration properties stored in non-volatile memory. These properties can be configured at the factory, by the key loader, or by the end user. More details about these properties can be found in a separate document which deals with all device functions other than key loading.

This device will go into suspend mode, and will wake up from suspend mode, when directed to do so by the host. This device does not support remote wakeup.

This device is powered from the USB bus. The vendor ID is 0x0801 and the product ID is 0x3004.

HID USAGES

HID devices send data in reports. Each report is identified by a unique identifier called a usage. The device's capabilities and the structure of its reports are sent 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. Vendor-defined usages must have a usage page in the range 0xFF00 – 0xFFFF. All usages for this device address vendor-defined IPAD usage page 0xFF20. The usage IDs for this device are defined in the following table, in which the usage types are also listed. These usage types are defined in the HID Usage Tables document.

Feature reports are used to send commands to the device and retrieve acknowledgement and data messages that are immediately available. Input reports are used by the device to send data to the host in an asynchronous manner when a related feature report completes or automatically when the device state changes.

REPORT DESCRIPTOR

The HID report descriptor is structured as follows: Note: this document relates only to the functions required for key injection. For a description of additional IPAD USB capabilities, see the IPAD Program Reference Manual, USB Communications, item number 99875430-1.

Item	Value (Hex)
Usage Page	06 20 FF
Usage (PINPAD)	09 01
Collection (Application)	A1 01
Report Size (8)	75 08
Logical Minimum (0)	15 00
Logical Maximum (255)	26 FF 00
Report ID (1)	85 01
Usage (Response ACK)	09 01
Report Count (4)	95 04
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (2)	85 02
Usage (End Session)	09 02
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (8)	85 08
Usage (Request Device Status)	09 08
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (9)	85 09
Usage (Get/Set Device Config)	09 09
Report Count (8)	95 08
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (11)	85 0B
Usage (Get/Set Challenge)	09 0B
Report Count (13)	95 0D
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (12)	85 0C
Usage (Set Bitmap)	09 0C
Report Count (2)	95 02
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (14)	85 0E
Usage (Get Information)	09 0E
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01

Report ID (16)	85 10
Usage (Send Big Block Data to Device)	09 10
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Item	Value (Hex)
Report ID (88)	85 58
Usage (Key Handling or Manufacturing Command)	09 58
Report Count (2)	95 02
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (32)	85 20
Usage (Device State)	09 20
Report Count (5)	95 05
Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Buf)	82 02 01
Report ID (41)	85 29
Usage (Send Big Block Data to Host)	09 29
Report Count (127)	95 7F
Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Buf)	82 02 01
End Collection	C0

FEATURE REPORTS

A number of feature reports have been defined in the IPAD to support data communications between the host and the device. Set feature is used by the host to send commands to the device. Get feature is used by the host to retrieve data or responses from the device.

Commands execute in the following sequence:

- Send feature report (command)
- Read feature report ID 0x01 (Response ACK) for acknowledgement, which includes the command number being acknowledged and one byte of status to indicate whether or not the command was accepted as sent
- (For some commands) Get feature reads data set up as a response to a command
- (For some commands) Input report response will be sent on the interrupt in pipe when a longer running command (e.g. Get Challenge) finishes

Feature Report List

Report ID (HEX)	Usage Name	Feature Type
01	Response ACK	Get Feature
02	End Session	Set Feature
08	Request Device Status	Set Feature
09	Set/Get Device Configuration	Get/Set Feature
0B	Get Challenge	Get/Set Feature
0C	Set Bitmap	Set Feature
0E	Get Key Information	Get Feature
10	Send Big Block Data to Device	Set Feature
58	Key Handling or Manufacturing Command	Set Feature

The generalized format of a feature report is as follows:

Bit	7	6	5	4	3	2	1	0
Byte 0	Report ID							
Byte 1	Data							
...	Data							

Report 0x01 - Response ACK

This command causes the IPAD to send the response status ("ACKSTS", see **APPENDIX A: STATUS AND MESSAGE TABLE**), and the Report ID of the command just executed, back to the host. The host should get this report immediately after it sends any command to the device to determine whether or not the device accepted the command as sent.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x01							
Byte 1	Status of Command ("ACKSTS")							
Byte 2	Report ID of Command being ACKd							

Report 0x02 – End Session

This command clears all existing session data including PIN, PAN, and amount. The device returns to the idle state and sets the display to the specified Welcome screen. Use of message IDs 1-4 require that the associated bitmaps have been previously loaded during configuration.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x02							
Byte 1	Idle message ID: 0 = Welcome (default) 1-4 : Use bitmaps (loaded as 0-3)							

Report 0x08 – Request Device Status

This command causes the IPAD to send current information (session state, device state and status, etc.) to the host via the interrupt in pipe. Following this command, the host should read an input report which contains the information (see **Report 0x20 – Device State Report**).

Bit	7	6	5	4	3	2	1	0
Byte 0	0x08							
Byte 1	0x00							

Report 0x09 – Set Device Configuration

Set feature 0x09 is used to send predefined (by user or host) device configuration data to the IPAD. If the current configuration is locked, then the device will report an error (0x87) in ACKSTS of **Report 0x01** and the new configuration will not be set. Otherwise, if the configuration data is OK, the new configuration will be saved.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x09							
Byte 1	Configuration	Bitmap	not defined				Require authentication	
	0 = unlocked 1 = locked	0 = unlocked 1 = locked					0 = no 1 = yes	
Byte 2	0x00							
Byte 3	Mask Configuration (default value = 0xC0, all enabled except MS2.0)							
	ISO Mask	Check Digit	00 = MS2.0 disabled 10 = MS2.0 enabled		Track 2 Data		Track 1 Data	
	0 = disabled 1 = enabled	0 = disabled 1 = enabled			Error	Blank	Error	Blank
Byte 4	MSR Card Configuration (default value = 0xD5, all enabled)							
	AAMVA Card	Non-finance card option	Track 3 Data		Track 2 Data		Track 1 Data	
	0 = disabled 1 = enabled		00 = disabled 01 = enabled 11 = required		00 = disabled 01 = enabled 11 = required		00 = disabled 01 = enabled 11 = required	
Byte 5	Mask Character							
Byte 6	Leading length to leave unmasked In MS2.0 format, if >8, set to 8; if <5, set to 5				Trailing length to leave unmasked Ignored in MS2.0 format			
Byte 7	0x00							
Byte 8	0x00							

Notes for Byte 3, bits 0 – 3:

- If Error = 0, build MS2.0 format Track data if at least one Track contains good data – the indicated Track number may contain error(s);
- If Error = 1, do not build MS2.0 format Track data if the indicated Track number contains error(s);
- If Blank = 0, build MS2.0 format Track data if at least one Track contains good data, – the indicated Track number may be blank;
- If Blank = 1, do not build MS2.0 format Track data if the indicated Track number is blank;

These four bits can contain any combination of values from 0000 to 1111.

Report 0x09 – Get Device Configuration

Get feature 0x09 will cause the IPAD to send the current device configuration to the host in the following report format:

Bit	7	6	5	4	3	2	1	0
Byte 0	0x09							
Byte 1	Configuration	Bitmap	not defined				Require authentication	
	0 = unlocked 1 = locked	0 = unlocked 1 = locked					0 = no 1 = yes	
Byte 2	0x00							
Byte 3	Mask Configuration (default value = 0xC0, all enabled except MS2.0)							
	ISO Mask	Check Digit	00 = MS2.0 disabled 10 = MS2.0 enabled		Track 2 Data		Track 1 Data	
	0 = disabled 1 = enabled	0 = disabled 1 = enabled			Error	Blank	Error	Blank
Byte 4	MSR Card Configuration (default value = 0xD5, all enabled)							
	AAMVA Card	Non-finance card option	Track 3 Data		Track 2 Data		Track 1 Data	
	0 = disabled 1 = enabled		00 = disabled 01 = enabled 11 = required		00 = disabled 01 = enabled 11 = required		00 = disabled 01 = enabled 11 = required	
Byte 5	Mask Character							
Byte 6	Leading length to leave unmasked In MS2.0 format, if >8, set to 8; if <5, set to 5				Trailing length to leave unmasked Ignored in MS2.0 format			
Byte 7	0x00							
Byte 8	0x00							

Report 0x0B – Get Challenge

This command causes the IPAD to send challenge information to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0B							
Byte 1	Key ID: 0x00 = PIN key 0x01 = MSR key 0x02 = PIN Cert 0x03 = MSR Cert 0x04 = Device Authentication signed by PIN cert 0x05 = Device Authentication signed by MSR cert 0x06 = Inject Fixed PIN key signed by PIN cert 0x08 = Inject Authentication key signed by PIN cert 0x09 = Inject Authentication key signed by MSR cert 0x10 = Inject Configuration signed by PIN cert 0x11 = Inject Configuration signed by MSR cert 0x63 = Authentication 0xFF = MFG command							

After sending this command to the device and getting the ACKSTS report, issue a Get Feature 0x0B for the Challenge Feature Report (see below). If the key ID is not in the list, or a valid authentication key is not available for key ID = 0x63, then the data block will be all zeros.

Challenge Feature Report

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0B							
Byte 1	Key ID: 0x00 = PIN key 0x01 = MSR key 0x02 = PIN Cert 0x03 = MSR Cert 0x04 = Device Authentication signed by PIN cert 0x05 = Device Authentication signed by MSR cert 0x06 = Inject Fixed PIN key signed by PIN cert 0x08 = Inject Authentication key signed by PIN cert 0x09 = Inject Authentication key signed by MSR cert 0x10 = Inject Configuration signed by PIN cert 0x11 = Inject Configuration signed by MSR cert 0x63 = Login/Logout/Authentication 0xFF = MFG command							
Byte 2	Data block:							
...	If Key_ID < 12 or Key_ID = 0xFF: Byte 2 – Byte 9 contains the device serial number Byte 10 – Byte 13 contains the random token							
Byte 13	If Key_ID = 0x63 and a valid authentication key is available: Byte 2 – Byte 9 contains the encrypted partial device serial number and random token Byte 10 – Byte 13 contains the partial device serial number							

Report 0x0C – Set Bitmap

This command causes the IPAD to save new bitmap image data in the specified slot with the selected format. The device can hold up to four different bitmaps in slots specified as 0-3. Slot 0 holds the default bitmap image.

In order to send new bitmap data to the IPAD, the following two steps are required:

- Issue **Report 0x10 – Send Big Block Data to Device** to send new bitmap image data to the device
- Issue **Report 0x0C – Set Bitmap** to request the device to save the new bitmap image data in the specified slot with the selected format

An error will be reported in ACKSTS of **Report 0x01** in the following cases:

- System is not available (0x8A)
- Bad parameters (0x82)
- Wrong Data Length (0x83)
- Bitmap configuration is locked (0x87)

If the flag is 0 (“clear”), then the current image will be cleared from the specified slot. Otherwise, if the command is successful, the new bitmap image data will be saved in the specified slot and will display (b/w or inverted) whenever the End Session command is invoked.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0C							
Byte 1	Bitmap Slot: Possible values: 0, 1, 2, 3							
Byte 2	Flag: 0 = clear, 1 = save, 2 = invert (i.e. reverse b/w) and save							

Report 0x0E – Get Information

This command causes the IPAD to send the requested information to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0E							
Byte 1	Info ID (see table of Info IDs and Data below)							

An error will be reported in ACKSTS of **Report 0x01** if the system is not available (0x8A) or the command contains bad parameters (0x82). Otherwise, the IPAD will send the following information feature report to the host:

Information Feature Report

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0E							
Byte 1	Info ID (see table of Info IDs and Data below)							
Byte 2	Key status, if Info ID < 0x80: 0 = Empty (default) 1 = OK 2 = Exhausted Key Status, if Info ID = 0x80: 0 – 5 = KCV type (see table of Info IDs and Data below)							
Byte 3	Data length (varies, see table of Info IDs and Data, below); default value is 0							
Byte 4	Block data							
...								

Table of Info IDs and Data

Info ID	Key Status	Data length	Data	Description
0x00	1	lflen*	Auth key label	If auth key exists
0x01,0x02	2	20	KSN	If no more keys
0x01	1	20	KSN	PIN key
0x02	1	20	KSN	MSR key
0x03	1	<=59	SN & subject's DN**	If PIN cert exists
0x04	1	<=59	SN & subject's DN**	If MSR cert exists
0x05	1	<=19	Label and KCV	If auth key exists
0x06	1	<=19	Label and KCV	If fixed key exists
0x10	1	4 x 3	4 slots for bitmap data [status + 2 bytes CRC] status: 0 = not loaded 1 = loaded	Bitmap data status and its CRC
0x50	1	9	Keypad sensitivity Tamper sensitivity Key on threshold Key off threshold 4 bytes keypad threshold Keypad calibration result	Keypad values
0x60 – 0x70	1	<=59	SN & subject's DN**	If associated CA cert exists***
0x71 – 0x7F	1	<=59	SN & issuer's DN**	If associated CA cert exists***
0x80	kcv_type=0	4	KCV value	KCV**** for Auth key
0x80	kcv_type=1	4	KCV value	KCV for PIN key
0x80	kcv_type=2	4	KCV value	KCV for MSR key
0x80	kcv_type=3	4	KCV value	KCV for fixed PIN key
0x80	kcv_type=4	4	Hash value	Dev auth key signed by PIN cert
0x80	kcv_type=5	4	Hash value	Dev auth key signed by MSR cert
0x80	All other kcv_types	0		KCV****

*: lflen = auth key's label length.

** : SN = serial number of cert;

DN = distinguished names of subject or issuer of cert;

Data length varies with SN and DN length; max length is 59.

***: its corresponding CA cert.

****: KCV = Key Check Value, where the lowest 6 digits are valid.

Report 0x10 – Send Big Block Data to Device

This command is used to provide data for **Report 0x58 – Key Handling or Manufacturing Command**, and **Report 0x0C – Set Bitmap**, in 60-byte increments. If the data size is greater than 60 bytes, then the data must be split into several small blocks, each containing a maximum of 60 bytes. Two data formats are used in connection with this command: the first packet (block 0) is used to signal the start of a new data set and to specify the complete length of the data; subsequent packets (blocks 1 through n) are used to transmit the actual data to a predefined buffer within the device.

An error will be reported in ACKSTS of **Report 0x01** in the following cases:

- The parameters in any block 1 through n data packet don't match (or don't follow) the previous data packet's parameters (0x82)
- Data length error (e.g. the data size is 0 or is larger than the available buffer size) (0x83)

Otherwise, if the command is successful, the bitmap image or key handling/manufacturing command will be stored in a predefined buffer within the device.

Start of Sending Format (Block 0)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x10							
Byte 1	Data type: 0x0C = Bitmap image data 0x58 = Key handling data or Manufacturing command							
Byte 2	0 = Start of new data set (this packet contains the total data length)							
Byte 3	Data length – low byte							
Byte 4	Data length – high byte							

Sending Data Format (Blocks 1 through n)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x10							
Byte 1	Data type: 0x0C = Bitmap image data 0x58 = Key handling data or Manufacturing command							
Byte 2	Data packet number (1..n)							
Byte 3	Packet length							
Byte 4	Packet data (If sending mfg, configuration, or key handling data, see table B below for details)							
...								
Byte 63								

Report 0x58 – Request Device Cert

This command causes the IPAD to send the device certificate to the host.

An error (0x80) will be reported in ACKSTS of **Report 0x01** if the IPAD detects a system error or finds that the device certificate is missing or has been changed. Otherwise, if the command is successful, input **Report 0x29 – Send Big Block Data to Host** will have been sent to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x58							
Byte 1	0x02							
Byte 2	0x01							
Byte 3	0x00							
...								
Byte 6								

Report 0x58 – Key Handling or Manufacturing Command

This command can be used to send the following key handling, configuration, or manufacturing commands to the device:

- Certificate
- Certificate Revocation List (CRL)
- Injection keys
- Bind/unbind/rebind
- Inject configuration
- Manufacturing command

An error will be reported in ACKSTS of **Report 0x01** in the following cases:

- Data size is greater than the predefined buffer size (0x83)
- A certificate or CRL has expired (0x91), is invalid (0x92), its associated CA doesn't exist (0x90) or has been revoked (0x93)
- MagTek IPAD OID of the certificate doesn't match the predefined OID (0x83)
- Message data (for item ID > 0x15) is not correct, including parameter data (0x82), its length (0x83), or signature (0x80)
- For bind/unbind/rebind/key injection, its corresponding CRL doesn't exist (0x94)
- For unbind/rebind/key injection, its associated certificate doesn't exist (0x90)
- For key injection, the key is poor (0x82) or the Auth/Fixed Key already exists (0x96)
- Tamper has been detected (0x80)

Otherwise, if the command completed successfully, then the corresponding operation will have already occurred (e.g., the key will have been sent to the device).

Because the data size is large, **Report 0x10 – Send Big Block Data to Device** is first used to send data to the device. After sending the data, issue the following command:

Bit	7	6	5	4	3	2	1	0
Byte 0	0x58							
Byte 1	0x01							
Byte 2	Item ID (see table A below)							

Table A: Load Cert/CRL/Key Item ID

Item ID	Description
0x00	Load MFG unbind cert
0x01	Load Device CA cert
0x02	Load PIN CA cert
0x03	Load MSR CA cert
0x04	Load Device cert
0x05	Bind PIN cert
0x06	Bind MSR cert
0x0F	Load PIN CRL
0x10	Load MSR CRL
0x15	Inject keys
0x16	Unbind PIN/MSR
0x17	Rebind PIN/MSR
0x18	MFG unbind PIN/MSR
0x19	MFG rebind PIN/MSR
0x1F	Inject configuration
0xFF	Manufacturing command

Table B: Block Data Format

Item ID	Description
0x00 – 0x06	DER form X509 certificate
0x0F – 0x10	DER form X509 CRL
0x15 – 0xFF	Message data block (see table C)

Table C: Message Data Block Format

Offset	Description
0	Signature length – low byte
1	Signature length – high byte
2	Msg length – low byte
3	Msg length – high byte
4	Header (see table D)
4 + header length	Msg (see table G)
4 + header length+ msg length	Signature (see table H)

*: To be signed message data is from offset 2 to the end of Msg.

Table D: Header Format

Offset	Description
0x00	Header length (1 byte)
0x01	Keyid (1 byte, see table E)
0x02	Device serial number (8 bytes)
0x0A	Random token (4 bytes)
0x0D	Extra field (varied length, see table F)

Table E: Keyid

Keyid	Description
0x00	PIN key for DUKPT key
0x01	MSR key for DUKPT key
0x02	PIN cert
0x03	MSR cert
0x04	Device Authentication signed by PIN cert
0x05	Device Authentication signed by MSR cert
0x06	Fixed PIN key signed by PIN cert*
0x08	Auth key signed with PIN cert*
0x09	Auth key signed with MSR cert*
0x0A	Configuration signed with PIN cert
0x0B	Configuration signed with MSR cert
0xFF	Manufacturing command

*: Auth/fixed PIN key cannot be the same and should satisfy odd parity check (see TDES key parity check for reference)

Table F: Extra Field

Item ID	Keyid	Data	Data length
0x15	0x00	KSN for DUKPT key	8 /16 bytes
0x15	0x01	KSN for DUKPT key	8 /16 bytes
0x15	0x04		0 bytes
0x15	0x05		0 bytes
0x15	0x06	Label for Fixed key	Varies
0x15	0x08	Label for Auth key	Varies
0x15	0x09	Label for Auth key	Varies
0x18	0x02 - 0x03	Serial number of cert	Varies
0x19	0x02 - 0x03	Serial number of cert	Varies

Table G: Msg Data

Item ID	Description
0x15	RSA-OAEP key encrypted with device public key
0x16	n/a
0x17	New cert
0x18	n/a
0x19	New cert
0x1F	Configuration (see Report 0x09 – Set Device Configuration)
0xFF	Manufacturing command (see table I)

Table H: Signature

Keyid	Item ID	Signed by
n/a	0x00	Device CA cert
n/a	0x01-0x03	Root cert
n/a	0x04	Device CA cert
n/a	0x05	PIN CA cert
n/a	0x06	MSR CA cert
n/a	0x0F	PIN CA cert
n/a	0x10	MSR CA cert
0x00	0x15-0x17	PIN cert
0x01	0x15-0x17	MSR cert
0x02-0x03	0x18-0x19	MFG unbind cert
0x04	0x15	PIN cert
0x05	0x15	MSR cert
0x06	0x15	PIN cert
0x08	0x15	PIN cert
0x09	0x15	MSR cert
0x0A	0x1F	PIN cert
0x0B	0x1F	MSR cert
0xFF	0xFF	MFG cert

*Signature scheme: RSA PKCS#1 V1.5 using hash function SHA1.

Table I: Manufacturing Command Data Format

Offset	Description
0	Mfg ID
1	Data block (varies, see table J)

Table J: MFG ID

MFG ID	Data block		DESCRIPTION
	Offset	Value	
5	n/a		Inject MAG head
6	0x02–0x1E	“Key” value	Active Keypad, “key” used to unlock the keypad
7	0x02	KeypadSensitivity	Keypad Calibration
	0x03	TamperSensitivity	
	0x04	Key-onThreshold	
	0x05	Key-offThreshold	
8	0x02–0x05	Tamper Threshold value	Set keyboard threshold
11	n/a		Activate tamper sensors
34	0x02	0 = 128-byte device key 1 = 256-byte device key	Get CSR
43	0x02	Month	Reset device clock to the host's current time. Year is from 2008.
	0x03	Day	
	0x04	Hour	
	0x05	Minute	
	0x06	Second	
	0x07	Wday	
	0x08	Year	

INPUT REPORTS

Input reports, which work as events, are data packets sent by the IPAD to the host via the USB Interrupt In pipe. Events occur when the device state changes or when an asynchronous command has completed.

Input Report List

Report ID (HEX)	Usage Name
0x20	Device State
0x29	Send Big Block Data to Host

Report 0x20 – Device State Report

This event is triggered explicitly when the host successfully issues **Report 0x08 – Request Device Status**, or automatically when the device changes state, either of which cause the IPAD to send Device State, Session State, Device Status, Device Certificate Status, and Hardware Status to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x20							
Byte 1	Device state (see APPENDIX A: STATUS AND MESSAGE TABLE)							
Byte 2	Session state (see APPENDIX A: STATUS AND MESSAGE TABLE)							
Byte 3	Device status (see APPENDIX A: STATUS AND MESSAGE TABLE)							
Byte 4	Device certificate status (see APPENDIX A: STATUS AND MESSAGE TABLE)							
Byte 5	Hardware status (see APPENDIX A: STATUS AND MESSAGE TABLE)							

Report 0x29 – Send Big Block Data to Host

This event is used to send the device certificate or CSR to the host upon successful completion of **Report 0x58 – Request Device Cert** or certain **Report 0x58 – Key Handling or Manufacturing Commands**. If the data size is greater than 123 bytes, the data must be broken into a few small data blocks, each having a maximum of 123 bytes. Three data formats are used in connection with this command:

- The first packet (block 0) is used to signal the start of sending, which defines the buffer type, buffer status, and the total length of data being sent (in bytes);
- Subsequent packets (blocks 1 through n) contain the requested data; and
- A final packet signifies the end of sending.

Start of Sending Format (Block 0)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x29							
Byte 1	big buffer type (0x02 = device cert, 0x42 = CSR)							
Byte 2	0x00 = start flag							
Byte 3	big buffer status (0x00 = N/A)							
Byte 4	data length–low byte							
Byte 5	data length–high byte							

Sending Data Format (Blocks 1 thru n)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x29							
Byte 1	not defined							
Byte 2	block number (options: 1 – 98)							
Byte 3	data length							
Byte 4	data block (maximum 123 bytes)							

End of Sending Format

Bit	7	6	5	4	3	2	1	0
Byte 0	0x29							
Byte 1	not defined							
Byte 2	99 = end flag							

APPENDIX A: STATUS AND MESSAGE TABLE

Status/Message	Value							
Operation status	0x00 = OK / Done 0x01 = User Cancel 0x02 = Timeout 0x03 = Host Cancel 0x04 = Verify fail 0x05 = Keypad Security							
ACK Status ("ACKSTS")	0x00 = OK / Done 0x80 = System Error 0x81 = System not Idle 0x82 = Data Error 0x83 = Length Error 0x84 = PAN Exists 0x85 = No Key or Key is incorrect 0x86 = System busy 0x87 = System Locked 0x88 = Auth required 0x89 = Bad Auth 0x8A = System not Available 0x8B = Amount Needed 0x90 = Cert non-exist 0x91 = Expired (Cert/CRL) 0x92 = Invalid (Cert/CRL/Message) 0x93 = Revoked (Cert/CRL) 0x94 = CRL non-exist 0x95 = Cert exists 0x96 = Duplicate KSN/Key							
Device State	0x00 = Idle 0x01 = Session 0x02 = Wait For Card 0x03 = Wait For PIN 0x04 = Wait For Selection 0x05 = Displaying Message 0x06 = Test (Reserved) 0x07 = Manual Card Entry 0x08 = Wait for Signature Capture							
Device Status	0x00 = OK Otherwise, the possible values are listed below, System – 1 = System Error (EndSession clears) Auth – 1 = Not Authorized (cleared when device is authenticated) Tamper – 1 = Tamper Detected MSR – 00 = OK – 01 = No MSR Key – 10 = MSR Key Exhausted – 11 = MSR Key not Bound PIN – 00 = OK – 01 = No PIN Key – 10 = PIN Key Exhausted – 11 = PIN Key not Bound							
	Bit 7	6	5	4	3	2	1	0
	System	Auth	0	Tamper	MSR		PIN	

Status/Message	Value							
Session State	The possible values are listed below, Pwr Chg – 1 = Power Change Occurred (occurs on Power up or after a USB resume) Card Data – 1 = Card Data Available MSR PAN – 1 = PAN Parsed from Card EXPAN – 1 = External PAN Sent Amt – 1 = Amount sent							
	Bit 7	6	5	4	3	2	1	0
	Pwr Chg	0	0	0	Card D a t a	MSRPAN	EXPAN	Amt
Device Certificate Status	0 = Certificate does not exist in the device 1 = Certificate exists in the device							
	Bit 7	6	5	4	3	2	1	0
	MSR CRL	PIN CRL	0	Mfg Unbind	MSR CA	PIN CA	Device CA	Device Cert
Hardware Status	0 = False 1 = True							
	Bit 7	6	5	4	3	2	1	0
	0	0	0	0	Keypad Activated	Keypad Calibrated	MagHead Pro-grammed	Tamper Sensors Active