IPAD

PROGRAMMING REFERENCE MANUAL
USB COMMUNICATIONS

PART NUMBER 99875430-7

JULY 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	24 Jun 09	Initial Release
2.01	12 Aug 09	Modified Manual Card Entry command; added screen shots
2.02	21 Sep 09	Corrected byte locations for commands 0x23, 0x24
2.03	6 Oct 09	Updated to correspond to the latest FW version, added screen shots
3.01	12 Jul10	Added documentation for MS2.0 formatting; modified description of CVC location in Manual Card Entry command; added description of the sig cap data output
4.01	25 Feb 11	Added new screenshot to Report 0x12; update Byte 2 options in report 0x06, 0x07, 0x11, 0x12; added example to report 0x12
5.01	4 May 2011	Updated to include options added in Rev C firmware
6.01	15 June 2011	Corrected Reports 0x10, 0x12 and 0x28
7.01	26 July 2011	Changed reference in Report 0x21 to 0x14 Request User Data Entry instead of 0x04 Request PIN Entry

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 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 later in this document in the command section and in a separate document which deals with 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:

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 (3)	85 03
Usage (Request Swipe Card)	09 03
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (4)	85 04
Usage (Request PIN Entry)	09 04
Report Count (5)	95 05
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (5)	85 05
Usage (Cancel Command)	09 05
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (6)	85 06
Usage (Request User Selection)	09 06
Report Count (4)	95 04
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (7)	85 07
Usage (Display Message)	09 07
Report Count (2)	95 02
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

Item	Value (Hex)
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 (10)	85 0A
Usage (Request MSR Data)	09 0A
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (11)	85 0B
Usage (Get 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 (13)	85 0D
Usage (Send Session Data)	09 0D
Report Count (21)	95 15
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 (15)	85 0F
Usage (Authenticate)	09 0F
Report Count (9)	95 09
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (16)	85 10
Usage (Send Big Block Data)	09 10
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (17)	85 11
Usage (Request Manual Card Entry)	09 11
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (18)	85 12
Usage (Request User Signature)	09 12
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01

Item	**Value (Hex)**
Report ID (19)	85 13
Usage (Get User Signature)	09 13
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (20)	85 14
Usage (Request User Data Entry)	09 14
Report Count (3)	95 03
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 (33)	85 21
Usage (User Data Entry Response)	09 21
Report Count (20)	95 14
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (34)	85 22
Usage (Card Status)	09 22
Report Count (16)	95 10
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (35)	85 23
Usage (Card Data)	09 23
Report Count (127)	95 7F
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (36)	85 24
Usage (PIN Response)	09 24
Report Count (20)	95 14
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (37)	85 25
Usage (User Selection Response)	09 25
Report Count (3)	95 03
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (39)	85 27
Usage (Display Message Done)	09 27
Report Count (2)	95 02
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (40)	85 28
Usage (Signature Capture State)	09 28
Report Count (4)	95 04
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01

Item	**Value (Hex)**
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

IPAD USB REPORTS

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) Read 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., Request PIN Entry or Request Swipe Card) finishes

Feature Report List

Report ID (HEX)	Usage Name	Feature Type
01	Response ACK	Get Feature
02	End Session	Set Feature
03	Request Swipe Card	Set Feature
04	Request PIN Entry	Set Feature
05	Cancel Command	Set Feature
06	Request User Selection	Set Feature
07	Display Message	Set Feature
08	Request Device Status	Set Feature
09	Set/Get Device Config	Get/Set Feature
0A	Request MSR Data	Set Feature
0B	Get Challenge	Get/Set Feature
0C	Set Bitmap	Set Feature
0D	Send Session Data	Set Feature
0E	Get Information	Get Feature
0F	Authenticate	Set Feature
10	Send Big Block Data to Device	Set Feature
11	Request Manual Card Entry	Set Feature
12	Request User Signature	Set Feature
13	Get User Signature	Get Feature
14	Request User Data Entry	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 Codes**), 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

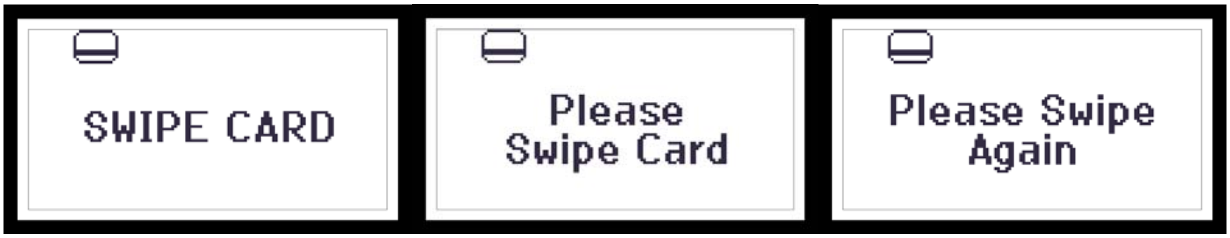
This command clears all existing session data including PIN, PAN, and amount. The device returns to the idle state and sets the display to the specified Welcome screen. Use of message IDs 1-4 require that the associated bitmaps have been previously loaded during configuration; otherwise, use 0 for displayMsg and the IPAD will display its default "Welcome" screen (shown below).

Welcome

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 0x03 – Request Swipe Card

This command causes the IPAD to prompt the user to swipe his or her card by displaying one of four predetermined messages (see Card Message ID, below); three examples are shown below:



SWIPE CARD

Please
Swipe Card

Please Swipe
Again

An error (in parentheses) will be reported in ACKSTS of **Report 0x01 – Response ACK** in the following cases:

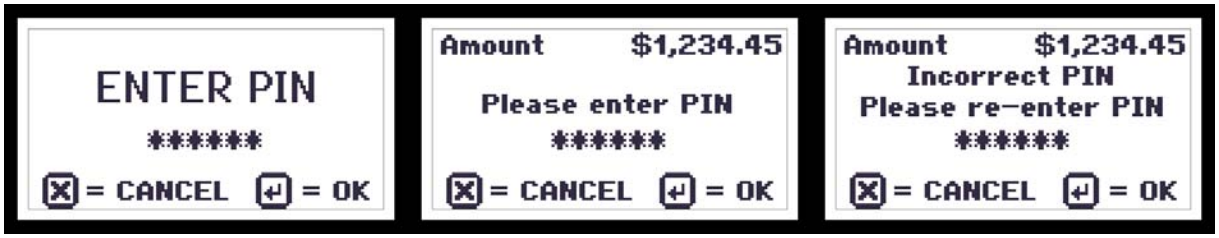
- System Error (0x80)
- System is not available (0x8A)
- Bad parameter (0x82)
- PAN already exists in the reader (0x84)

When this command completes (card swiped OK, user cancelled, or timeout), the device will send of **Report 0x22 – Card Status Report** to the host. If the Card and Operation Status are both OK, then the host should send a request to get the card data (see **Report 0x0A – Request MSR Data**).

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x03							
Byte 1	Wait time in seconds, (1 – 255; 0 = infinite wait time)							
Byte 2	Card Message ID to display: 0 = Swipe Card / Idle alternating 1 = Swipe Card 2 = Please Swipe Card 3 = Please Swipe Card Again							
Byte 3	Tones: 0 = No sound 1 = One beep 2 = Two beeps							

Report 0x04 – Request PIN Entry

This command causes the IPAD to prompt the user to enter his or her PIN by displaying one of five predetermined messages (see PIN Mode, below); three examples are shown below:



ENTER PIN
******
☒ = CANCEL ↵ = OK

Amount $1,234.45
Please enter PIN
******
☒ = CANCEL ↵ = OK

Amount $1,234.45
Incorrect PIN
Please re–enter PIN
******
☒ = CANCEL ↵ = OK

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

- Bad parameter (0x82)
- System is locked (more than 120 PINs were entered within one hour) (0x87)
- System is not available (0x8A)
- If PIN amount is required, no amount has been sent (0x8B)

Otherwise, when the command completes (PIN entry done, user cancelled, or timeout), the IPAD will send **Report 0x24 – PIN Response Report** to the host by interrupt in pipe. If PIN entry is successful, the report will also contain the PIN KSN (if using a DUKPT PIN Key, otherwise the PIN KSN will be zero) and the encrypted PIN block (EPB) data. The EPB format will depend on the PIN option and Session State. If there is no PAN (from card swipe or sent via command), then the EPB will use ISO format 1. If a PAN exists, then the PIN option will be used to determine if the created PIN block will be ISO format 0 (for VerifyPin) or ISO format 3. If the VerifyPIN option is set, the IPAD will request the user to enter his or her PIN twice and will generate an EPB only if both entries match. The EPB is encrypted under the current PIN DUKPT key as DES or TDES depending on the injected key type. The WaitMsg option will cause the device to display a **Please Wait** message during the delay (the unit is checking for keypad tamper) before the **Enter PIN** message is displayed.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x04							
Byte 1	Wait Time in seconds, (1 – 255; 0 = 256 seconds)							
Byte 2	PIN Mode: 0 = Enter Pin 1 = Enter Pin Amount 2 = Reenter PIN Amount 3 = Reenter PIN 4 = Verify PIN							
Byte 3	Max PIN length (<= 12)				Min PIN length (>=4)			
Byte 4	Tones: 0 = No sound 1 = One beep 2 = Two beeps							
Byte 5	PIN options							
						Wait Msg	VerifyPIN	ISO3

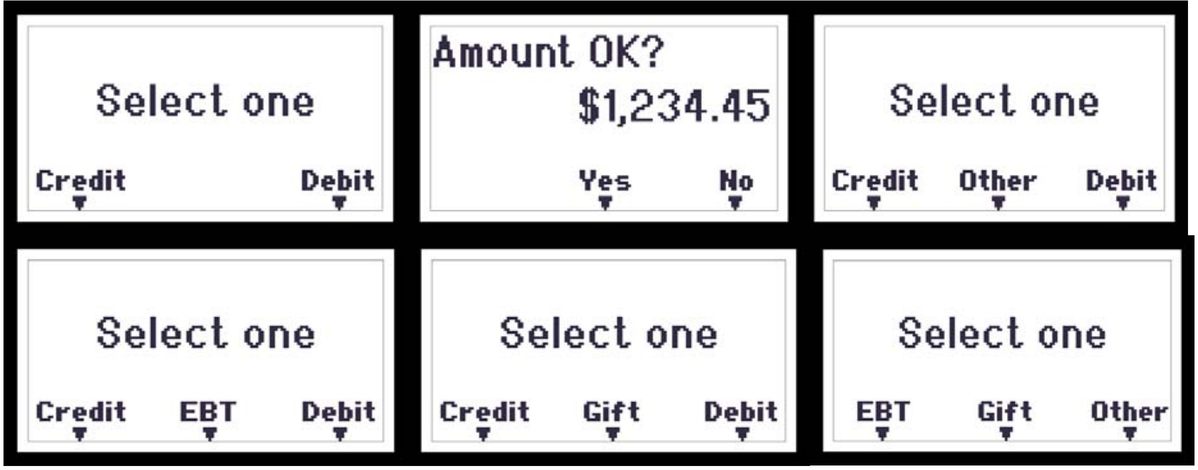
Report 0x05 – Cancel Command

This command is used to cancel the current command.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x05							
Byte 1	0							

Report 0x06 – Request User Selection

This command causes the IPAD to prompt the user to select the transaction type (credit, debit, gift, ebt, or other), or to verify the transaction amount, as shown below:



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

- System is not available (0x8A)
- Bad parameter (0x82)
- If transaction amount is required, no amount has been sent (0x8B)

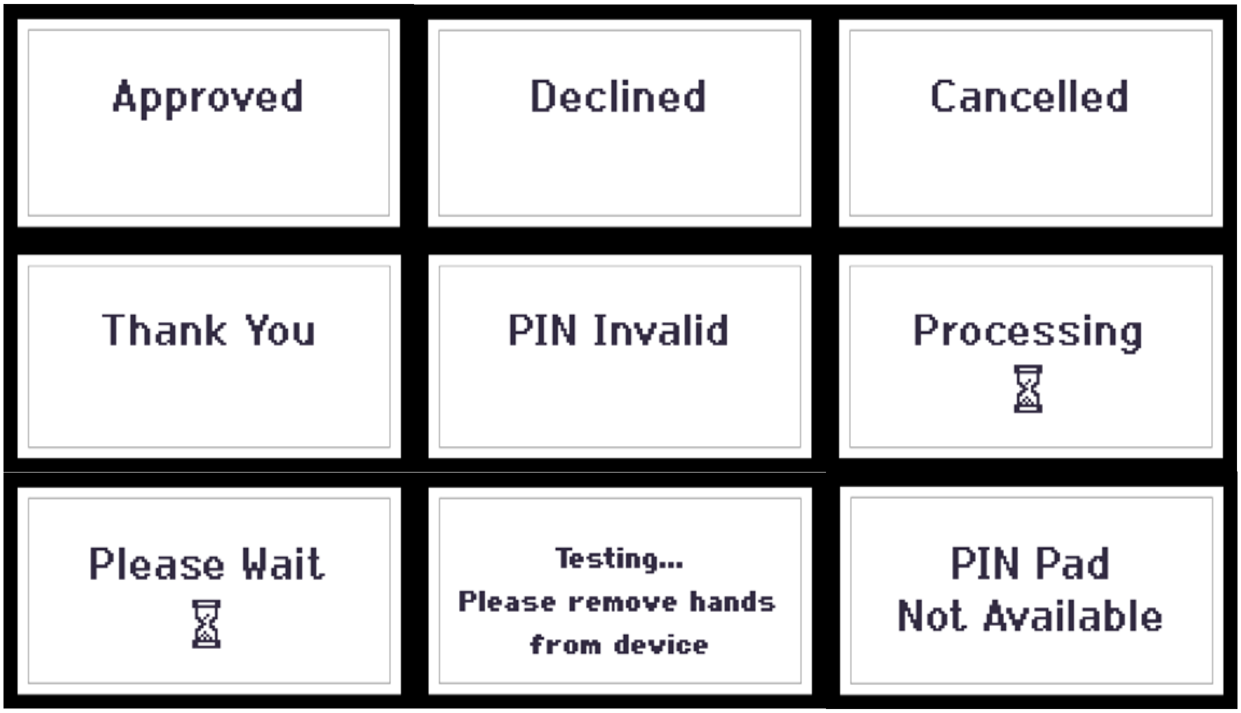
Otherwise, when the command completes (selection made, user cancelled, or timeout):

- The LCD will be cleared
- The device will return to the idle state
- **Report 0x25 – User Selection Response Report** will be sent to the host

Bit	7	6	5	4	3	2	1	0
Byte 0	0x06							
Byte 1	Wait Time in seconds, (1 – 255; 0 = 256 seconds)							
Byte 2	Message ID: 0 = Transaction Type (credit/debit) 1 = Verify Transaction Amount 2 = Transaction Type (credit//other/debit) 3 = Transaction Type (credit/ebt/debit) 4 = Transaction Type (credit/gift/debit) 5 = Transaction Type (ebt/gift/other) 255 = User (requires first sending data via Report 0x10 – Send Big Block Data to Device)							
Byte 3	Mask Key:							
					Enter	Right	Middle	Left
Byte 4	Tones: 0 = No sound 1 = One beep 2 = Two beeps							

Report 0x07 – Display Message

This command causes the IPAD to display one of nine predefined messages on its LCD for a specified time, as shown below:



Approved

Declined

Cancelled

Thank You

PIN Invalid

Processing

Please Wait

Testing...
Please remove hands
from device

PIN Pad
Not Available

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

- Bad parameter (0x82)
- System is not available (0x8A)

Otherwise, when the command completes (message displayed, user cancelled, or timeout):

- The LCD will be cleared
- The device will return to the idle state
- **Report 0x27 – Display Message Done Report** will be sent to the host

Bit	7	6	5	4	3	2	1	0
Byte 0	0x07							
Byte 1	Wait Time in seconds, (1 – 255; 0 = infinite wait time)							
Byte 2	Display message ID: 0 – Blank 1 – Approved 2 – Declined 3 – Cancelled 4 – Thank You 5 – PIN Invalid 6 – Processing 7 – Please Wait 8 – Hands Off 9 – PIN PAD not available 128-131 = Bitmap in slots 0-3 255 = User (requires first sending data via Report 0x10 – Send Big Block Data to Device)							

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) configuration data to the IPAD. If the current configuration is locked, then the device will report an error (0x87) in ACKSTS of **Report 0x01 – Response ACK** 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 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 0x0A – Request MSR Data

This command causes the IPAD to send MSR data to the host; therefore, it should be issued after a **Report 0x03 – Request Swipe Card** or **Report 0x11 – Request Manual Card Entry** command has successfully completed. If the system is not available, then the device will report an error (0x8A) in ACKSTS of **Report 0x01 – Response ACK**. Otherwise, the device will send multiple **Report 0x23 – Card Data Reports** to the host. Note: if no MSR data is available, then the device will send a single Report 23 containing a Data Length of 0.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0A							
Byte 1	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: 0x63 = Authentication							

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: 0x63 = Login/Logout/Authentication							
Byte 2 Byte 13	Data block: If 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 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 – Response ACK** in the following cases:

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

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 stored in the specified slot with the selected format, and will display whenever the End Session command is invoked.

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

Report 0x0D – Send Session Data (Amount)

This command is used to send transaction data (credit or debit card amount) to the device.

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

- Data error (0x82)
- Wrong data length (0x83)
- System is not available (0x8A)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0D							
Byte 1	0x00							
Byte 2	Amount length: 1 -11							
Byte 3	Reserved for future use							
Byte 4 …	Amount data in ASCII format							

Report 0x0D – Send Session Data (PAN)

This command is used to send card PAN data to the device.

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

- Data error (0x82)
- Wrong data length (0x83)
- The PAN already exists (0x84)
- System is not available (0x8A)

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0D							
Byte 1	0x01							
Byte 2	PAN data length: 8-19							
Byte 3	PAN data in ASCII format							

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 – Response ACK** 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	lbllen*	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
0x11	1	16	Flash signature data	Flash signature
0x50	1	8	Keypad sensitivity Tamper sensitivity Key on threshold Key off threshold 4 bytes keypad threshold	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****

*: lbllen = 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 0x0F – Login/Authenticate

This command logs in the device.

The following steps are required before issuing this command:

- Host requests an authentication token from the device (using **Report 0x0B – Get Challenge**)
- Host decrypts the received token with the authentication key
- Host transforms token and encrypts it with the authentication key

Authentication will fail, and an error will be reported in ACKSTS of **Report 0x01 – Response ACK,** in the following cases:

- System Error (e.g., a system error or tamper has been detected) (0x80)
- No authentication key is found in the device (0x85)
- Authentication is locked out (occurs after 3 authentication failures) (0x87)
- Host receives an incorrect authentication token (e.g., the decrypted random token or device serial number doesn't match the device's current values) (0x89)
- Authentication challenge token times out (i.e. is not used within 5 minutes) (0x8A)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x0F							
Byte 1	0x01 = Login/Authenticate							
Byte 2	Encrypted random token and device serial number (8 bytes) (see **Report 0x0B – Get Challenge**)							

Report 0x0F – Logout

This command logs out the device.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x0F							
Byte 1	0x00 = logout							

Report 0x10 – Send Big Block Data to the Device

This command is used to provide data for **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 buffer within the device.

An error will be reported in ACKSTS of **Report 0x01 – Response ACK** 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 sucessful, the bitmap image data 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							
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							
Byte 2	Data packet number (1..n)							
Byte 3	Packet length							
Byte 4 … Byte 63	Packet data							

User screen control consists of a data block constructed as described below and sent using Report 0x10 – Send Big Block Data to Device.

Byte 1 = number of strings

Each string consists of:

1 byte length (including params)
1 byte X location (0-127)
1 byte Y location (0-63)
1 byte P1 [t u pp aa ff]

ff – font – 0 = small, 1 = small bold, 2 = big
aa – align – 0 = left, 1 = center, 2 = right
pp – spacing – 0=proportional, 1 = proportional except numbers, 2 = fixed spacing
u – 1 = underline
t – 0 = transparent (background unchanged), 1 = background near characters is cleared

1 byte P2 (reserved – set to 0)
String as byte array excluding any zero terminator

Example (used for the user select command):

```
MemoryStream ms = new MemoryStream();
ms.WriteByte(4);  // # of strings
addUserString(ms, 19, 56, 0x25, 0, "$20");
addUserString(ms, 64, 56, 0x25, 0, "$40");
addUserString(ms, 112, 56, 0x25, 0, "$100");
addUserString(ms, 64, 30, 0x15, 0, "Select Cashback");
pp.SendMultiData(6, ms.ToArray());  //6 for getsel, 7 for
disp
pp.GetResponse(30, ResponseMsg.UserMsg, KeyMask.Left |
KeyMask.Right | KeyMask.Middle, 0);
```

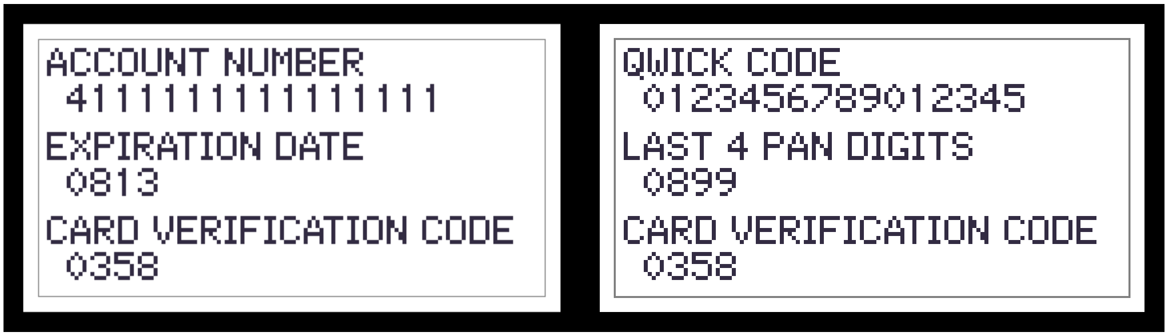
Report 0x11 – Request Manual Card Entry

This command causes the IPAD to prompt the user to enter the following Card information by keypad in the screen shown below:

1. Account number (mininum length = 9, maximum length = 19)
2. Expiration date (mininum length = maximum length = 4)
3. Card verification code (mininum length = 3, maximum length = 4)

Or

1. Qwick Code (mininum length = 8, maximum length = 16)
2. Last 4 digits of account # (mininum length = maximum length = 4)
3. Card verification code (mininum length = 3, maximum length = 4)



ACCOUNT NUMBER
4111111111111111
EXPIRATION DATE
0813
CARD VERIFICATION CODE
0358

QWICK CODE
0123456789012345
LAST 4 PAN DIGITS
0899
CARD VERIFICATION CODE
0358

An error will be reported in ACKSTS of **Report 0x01 – Response ACK** if the Device Status is not OK (0x8A).

When this command completes, **Report 0x22 – Card Status Report** will be sent back to the host. If the host or user canceled the request, or the request timed out, then byte 1 of **Report 0x22 – Card Status Report** will contain the appropriate Operation Status code to indicate why this command did not complete. Otherwise, if all of the card information was entered correctly, then byte 1 = 0x00 (this command completed OK), byte 2 = 0x00 (Card Status is OK), byte 3 = 0x03 (Card Type is manual), and the host should send a request to get the card data (see **Report 0x0A – Request MSR Data**). If Card and Operation Status are both OK, then the host should send a request to get the card data. **Report 0x20 – Device State Report** will also be sent back to update the current Device State.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x11							
Byte 1	Wait Time in seconds, (1 – 255; 0 = 256 seconds)							
Byte 2	0				Set to 1 to use PAN in PIN block creation	Set to 1 for Qwick Codes entry.	Field Options 0 = Acct,Date,CVC 1 = Acct,Date 2 = Acct,CVC 3 = Acct	
Byte 3	Tones: 0 = No sound 1 = One beep 2 = Two beeps							

Track data formatting for card data manually entered:

The track data sent by the IPAD for manually entered card data may be masked according to the IPAD's configuration (the same as it is for credit/debit cards), but the data shown in the following examples is unmasked just to show the detail. The account number (or QwickCode) is denoted by a string of 5s, the expiration date (or PAN4) by 3s and the CVC by 4s. The location marked by '6' will indicate the field options used when the data was collected – unused fields will be 0s. 0's below denote fixed-length filler. Track 1 card type ('B' for credit/debit cards) is set to 'M' and the name is set to the literal "MANUAL ENTRY/".

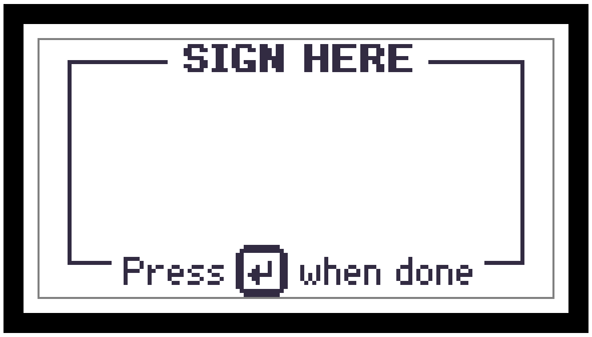
Track 1 data may be found in the Card Report (see **Report 0x23 – Card Data Reports**) that contains Data ID = 0x01. The IPAD will format Track 1 card data as follows:
%M5555555555555555^MANUAL ENTRY/^33330000004444000006?

Track 2 data may be found in the Card Report (see **Report 0x23 – Card Data Reports**) that contains Data ID = 0x02. The IPAD will format Track 2 card data as follows:
;5555555555555555=33330000004444006?

Note: The IPAD does not change the length of the CVC (either 3 or 4 characters) entered by the user. The length of the CVC thus affects the length of the Track data output by the IPAD, and the host must locate the CVC in the Track data as follows: The CVC starting position is the byte after the 6 digits which follow the 4-digit expiration date (or PAN4). The CVC ending position in Track 1 is the byte before the 6 digits which precede the end sentinel (?); the CVC ending position in Track 2 is the byte before the 3 digits which precede the end sentinel (?).

Report 0x12 – Request User Signature

This command causes the IPAD to request the user's signature in the screen shown below:



SIGN HERE

Press ↵ when done

An error (0x8A) will be reported in ACKSTS of **Report 0x01 – Response ACK** if the system is not available or the Touch Screen is not connected or doesn't exist.

Otherwise, when this command completes, **Report 0x28 – Signature Capture State Report** will be sent back to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x12							
Byte 1	Wait Time in seconds, (1 – 255; 0 = 256 seconds)							
Byte 2	Options: 0 = Timeout clears any signature data 1 = Timeout returns timeout status plus length collected. Sig Data can be requested.							
Byte 3	Tones: (optional) 0 = No sound 1 = One beep 2 = Two beeps							

Report 0x13 – Get User Signature

This command causes the device to send the user's signature data to the host.

An error (0x8A) will be reported in ACKSTS of **Report 0x01 – Response ACK** if the system is not available or the Touch Screen is not connected or doesn't exist.

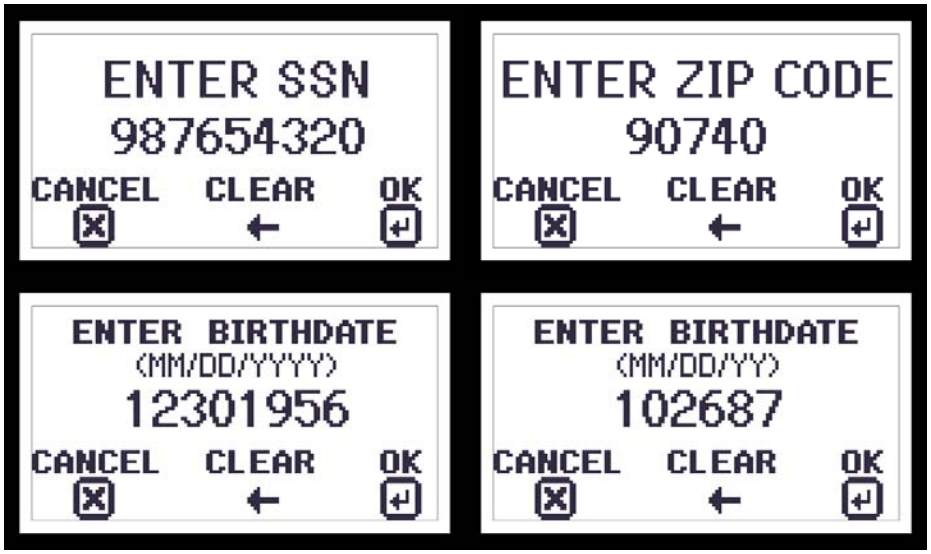
Otherwise, when this command completes, **Report 0x29 – Send Big Block Data to Host,** which contains the user's signature data, will be sent back to the host.

The user's signature data is a block of contiguous two-byte Hexadecimal pairs defining points (e.g., X1,Y1,X2,Y2,X3,Y3…), where X can range from 0-255 on the x axis, and Y can range from 0-127 on the Y axis. Y can also be 255, which represents a pen lift up. For example, if the User signed with an "X", the data might appear as 050A0A0500FF0A0A050500FF.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x13							

Report 0x14 – Request User Data Entry

This command causes the IPAD to prompt the user to enter his or her SSN, Zip code, or Birth date by displaying one of four predetermined messages, examples of which are shown below:



ENTER SSN
987654320

ENTER ZIP CODE
90740

ENTER BIRTHDATE
(MM/DD/YYYY)
12301956

ENTER BIRTHDATE
(MM/DD/YY)
102687

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

- Bad parameter (0x82)
- System is not available (0x8A)

Otherwise, when the command completes (data entry done, user cancelled, or timeout), the IPAD will send **Report 0x21 – User Data Entry Response Report** to the host by interrupt in pipe. If data entry is successful, the report will also contain the MSR KSN and the encrypted user data block (EUDB). The EUDB format is similar to the PIN ISO format 1 data block. The EUDB is encrypted using X9.24 data variant under the current data variant derived from the MSR key.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x14							
Byte 1	Wait Time in seconds, (1 – 255; 0 = 256 seconds)							
Byte 2	User data Mode: 0 = Enter SSN (9 digits) 1 = Enter Zip code (5 digits) 2 = Enter Birthdate (8 digits, in MM/DD/YYYY format) 3 = Enter Birthdate (6 digits, in MM/DD/YY format)							
Byte 3	Tones: 0 = No sound 1 = One beep 2 = Two beeps							

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
0x21	User Data Entry Response
0x22	Card Status
0x23	Card Data
0x24	PIN Response
0x25	User Selection Response
0x27	Display Message Done
0x28	Signature Capture 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, and Device 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 Codes)**							
Byte 2	Session State (see **Appendix A. Status and Message Codes**)							
Byte 3	Device Status (see **Appendix A. Status and Message Codes**)							
Byte 4	Device Certificate Status (see Appendix A. Status and Message Codes)							
Byte 5	Hardware Status (see Appendix A. Status and Message Codes)							

Report 0x21 – User Data Entry Response Report

This event is triggered by **Report 0x14 – Request User Data Entry**, which causes the IPAD to send User data to the host after the user has successfully entered data.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x21							
Byte 1	Operation Status (see **Appendix A. Status and Message Codes**)							
Bytes 2-11	MSR KSN							
Bytes 12-19	Encrypted User Data block							

Raw User Data Structure

a. SSN (9 digits)

Bits	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	C	**N**	**P**	**P**	**P**	**P**	**P**	**P**	**P**	**P**	**P**	**R**	**R**	**R**	**R**	**R**

b. Zip code (5 digits)

Bits	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	C	**N**	**P**	**P**	**P**	**P**	**P**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**

c. Birth Date (8/6 digits: mmddyyyy/mmddyy format)

Bits	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
	C	**N**	**P**	**P**	**P**	**P**	**P**	**P**	**P/R***	**P/R***	**R**	**R**	**R**	**R**	**R**	**R**

Where: C: control field (0100=SSN; 0101=Zip Code; 0110=Birth Date)
N: the data length
P: user data digit from 0000 (decimal 0) to 1001 (decimal 9)
R: filled random number
* Note: if the Birth Date data length is 6 (mmddyy format), then these positions will be filled with random numbers; if the Birth Date data length is 8 (mmddyyyy format), then these positions will contain the rightmost two characters of the Birth year.

Report 0x22 – Card Status Report

This event is triggered by **Report 0x03 – Request Swipe Card**, or by **Report 0x11 – Request Manual Card Entry**, either of which cause the IPAD to send Operation Status, Card Status, and Card Type to the host.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x22							
Byte 1	Operation Status (see **Appendix A. Status and Message Codes**)							
Byte 2	Card Status (see **Appendix A. Status and Message Codes**)							
Byte 3	Card Type (see **Appendix A. Status and Message Codes**)							

Report 0x23 – Card Data Reports

This event is triggered by **Report 0x0A – Request MSR Data**, which causes the IPAD to send eight reports to the host for each successful card swipe or manual card entry.

Bit	7	6	5	4	3	2	1	0
Byte 0	0x23							
Byte 1	Data ID: 0x01 = Track 1 data 0x02 = Track 2 data 0x03 = Track 3 data 0x04 = Encrypted Track 1 data 0x05 = Encrypted Track 2 data 0x06 = Encrypted Track 3 data 0x07 = Encrypted MagnePrint data 0x63 = KSN and MagnePrint Status							
Byte 2	Track Status: 0x00 = OK 0x01 = Empty 0x02 = Error 0x03 = Disabled							
Byte 3	Data length							
Byte 4 ...	Data block If Data ID < 0x08, data is track, encrypted track, or MP data corresponding to its data ID If Data ID = 0x63, Byte 4 -13 is 10 bytes KSN data, Byte 14-17 is 4 bytes MP Status data							

For MS2.0 format, track status (byte 2) of report 0x63 can be used for MS2.0 format status, from 0x00 to 0x15, which is defined as:

MS2.0 format status code

value	comment
0x00	SUCCESS
0x01	N/A
0x02	NO_TK2_FS
0x03	BAD_TK2_PAN_LEN
0x04	NO_FIRST_TK1_FS
0x05	NO_SECOND_TK1_FS
0x06	NO_TK1_ES
0x07	NO_TK2_ES
0x08	TK1_TRAIL_TOO_SHORT
0x09	TK1_AND_TK2_PANS_NOT_EQUAL
0x0A	BAD_TK1_FC
0x0B	DATA_NOT_ASCII_DECIMAL
0x0C	BAD_TK2_PAN_PREFIX
0x0D	BAD_ADDITIONAL_DATA
0x0E	TK1_LEN_TOO_LONG
0x0F	DATA_PROHIBITED_CHARS
0x10	TK1_BLANK
0x11	TK1_ERROR
0x12	TK2_BLANK
0x13	TK2_ERROR
0x14	NOTRACKDATA
0x15	TK1_PANTOOSHORT

Report 0x24 – PIN Response Report

This event is triggered by **Report 0x04 – Request PIN Entry**, which causes the IPAD to send PIN data to the host after a PIN is successfully entered.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x24							
Byte 1	Operation Status (see **Appendix A. Status and Message Codes**)							
Bytes 2-11	PIN KSN. If fixed PIN key is used, then KSN is zero.							
Bytes 12-19	Encrypted PIN block							

Report 0x25 – User Selection Response Report

This event is triggered by **Report 0x06 – Request User Selection,** which causes the IPAD to send the user's response (i.e. the key pressed) to the host.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x25							
Byte 1	Operation Status (see **Appendix A. Status and Message Codes**)							
Byte 2	Code of Key Pressed							

Report 0x27 – Display Message Done Report

This event is triggered by **Report 0x07 – Display Message**, which causes the IPAD to send a status report to the host to indicate that the previous **Report 0x07 – Display Message** has completed successfully.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x27							
Byte 1	Operation Status							

Report 0x28 – Signature Capture State Report

This event is triggered by **Report 0x12 – Request User Signature**, which causes the IPAD to send a status report to the host to indicate that the previous **Report 0x12 – Request User Signature** has completed successfully.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 0	0x28							
Byte 1	Operation Status							
Byte 2	0x00 (rfu)							
Byte 3	Signature length (low byte)							
Byte 4	Signature length (high byte)							

Report 0x29 – Send Big Block Data to Host

This event is used to send the user's signature to the host upon successful completion of **Report 0x13 – Get User Signature**. 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 (0x00 = signature capture)							
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							

EXAMPLES

How to get MSR and PIN data from the device for use with a bank simulation program?
(Note: all data shown in this section is in hex format)

1) Host sends out **Report 0x03 – Request Swipe Card** to the device.

   Sample command data of **Report 0x03 – Request Swipe Card**:
   03 20 00 01

   03 : report ID (03=**Report 0x03 – Request Swipe Card**)
   20 : wait time (20=32 seconds)
   00 : display message ID (00=swipe card/idle)
   01 : beep prompt tone for card swipe (01=one beep)

2) Device sends back **Report 0x01 – Response ACK** to host.

   Sample of response for **Report 0x01 – Response ACK**
   01 00 03

   01 : report ID (01=**Report 0x01 – Response ACK**)
   00 : ACK status of **Report 0x03 – Request Swipe Card** (00=command is good)
   03 : report ID of command ACKd (03=**Report 0x03 – Request Swipe Card**)

   Note: if the **Report 0x03 – Request Swipe Card** command failed (i.e. ACK status not = 00), then the IPAD will not return a device state input report to the host.

3) The device will prompt the user to swipe his or her card and will also send **Report 0x20 – Device State Report** to the host.

   Sample of device state input report
   20 02 08 40 47 07

   20 : report ID (20=**Report 0x20 – Device State Report**)
   02 : device state (02=wait for card)
   08 : session state (08=card data available)
   40 : device status (40=not authenticated)
   47 : Reserved for future use
   07 : Reserved for future use

4) After the card is swiped, the device will send back **Report 0x22 – Card Status Report** to the host.

   Sample data of card status input report:
   22 00 00 01

   22 : report ID (22 = **Report 0x22 – Card Status Report**)
   00 : operation status (00=OK)
   00 : card status (00=OK)
   01 : card type (01=financial card)

5) If both operation and card status are OK, then the host will retrieve the card data from the device by issuing **Report 0x0A – Request MSR Data**.

   Sample data of **Report 0x0A – Request MSR Data:**
   0A 00

6) The device will send back **Report 0x01 – Response ACK** to the host.

7) The device will send back eight **Report 0x23 – Card Data Reports** to the host.

   Sample card data:

   Track 1: 23 01 00 2F 0-0x2E bytes of data
   Track 2: 23 02 00 1E 0-0x1D bytes of data
   Track 3: 23 03 00 47 0-0x46 bytes of data
   Encrypted Track1: 23 04 00 30 0-0x2F bytes of data
   Encrypted Track2: 23 05 00 20 0-0x1F bytes of data
   Encrypted Track3: 23 06 00 48 0-0x47 bytes of data
   Encrypted MagnePrint: 23 07 00 38 0-0x37 bytes of data
   KSN and MagnePrintStatus: 23 63 00 0E 0-0x0D bytes of data

8) The device will send back another **Report 0x20 – Device State Report** to the host.

   If the operation status and card status from **Report 0x22 – Card Status Report** are both OK, the host shall issue **Report 0x04 – Request PIN Entry**

   Sample data:
   04 1E 00 44 01 01

   04 : report id (04=**Report 0x04 – Request PIN Entry)**
   1E : wait time for PIN entry (1E=30 seconds)
   00 : PIN mode (00=enter PIN)
   44 : Max and Min length of PIN (in this example, PIN must be exactly four characters)
   01 : prompt tone (01=one beep)
   01 : PIN option (01=ISO3)

9) The device will send back **Report 0x01 – Response ACK** if the command is successful.

10) The device will send back **Report 0x24 – PIN Response Report** if PIN entry is sucessful.

11) The device will send back another **Report 0x20 – Device State Report** to the host.

APPENDIX A. STATUS AND MESSAGE CODES

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
Display Message	0x00 = Hands Off 0x01 = Approved 0x02 = Declined 0x03 = Cancelled 0x04 = Thank You 0x05 = PIN Invalid 0x06 = Processing 0x07 = Please Wait
Function Key	0x71 = Left 0x72 = Middle 0x74 = Right 0x78 = Enter
Pin Message	0x00 = Enter Pin 0x01 = Enter Pin Amount 0x02 = Reenter PIN Amount 0x03 = Reenter PIN 0x04 = Verify PIN
Response Message	0x00 = TransactionType 0x01 = Amount OK
Card Message	0x00 = Swipe Card / Idle alternating 0x01 = SwipeCard 0x02 = Please Swipe Card 0x03 = Please Swipe Again
Buzzer	0x00 = None 0x01 = Single Beep 0x02 = Double Beep
Amount Type	0x00 = Credit 0x01 = Debit

<table>
<tr><th>Status/Message</th><th colspan="8">Value</th></tr>
<tr><td>Device State</td><td colspan="8">0x00 = Idle
0x01 = Session
0x02 = Wait For Card
0x03 = Wait For PIN
0x04 = Wait For Selection
0x05 = Displaying Message
0x06 = Test (Reserved for future use)
0x07 = Manual Card Entry
0x08 = Wait for Signature Capture</td></tr>
<tr><td>Card Type</td><td colspan="8">0x00 = Other
0x01 = Financial
0x02 = AAMVA
0x03 = Manual
0x04 = Unknown</td></tr>
<tr><td rowspan="3">Card Status</td><td colspan="8">0x00 = OK
Otherwise, for each track, the possible values are listed below:
Value 0 = no error
Value 1 = error detected</td></tr>
<tr><td><b>Bit 7</b></td><td><b>6</b></td><td><b>5</b></td><td><b>4</b></td><td><b>3</b></td><td><b>2</b></td><td><b>1</b></td><td><b>0</b></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Track 3</td><td>Track 2</td><td>Track 1</td><td>0</td></tr>
<tr><td rowspan="3">Key Mask</td><td colspan="8">For each key, the possible values are listed below:
Value 0 = the indicated key was not pressed
Value 1 = the indicated key was pressed</td></tr>
<tr><td><b>Bit 7</b></td><td><b>6</b></td><td><b>5</b></td><td><b>4</b></td><td><b>3</b></td><td><b>2</b></td><td><b>1</b></td><td><b>0</b></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Enter</td><td>Right</td><td>Middle</td><td>Left</td></tr>
<tr><td rowspan="3">Device Status</td><td colspan="8">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</td></tr>
<tr><td><b>Bit 7</b></td><td><b>6</b></td><td><b>5</b></td><td><b>4</b></td><td><b>3</b></td><td><b>2</b></td><td><b>1</b></td><td><b>0</b></td></tr>
<tr><td>System</td><td>Auth</td><td>0</td><td>Tamper</td><td colspan="2">MSR</td><td colspan="2">PIN</td></tr>
<tr><td rowspan="3">Session State</td><td colspan="8">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</td></tr>
<tr><td><b>Bit 7</b></td><td><b>6</b></td><td><b>5</b></td><td><b>4</b></td><td><b>3</b></td><td><b>2</b></td><td><b>1</b></td><td><b>0</b></td></tr>
<tr><td>Pwr Chg</td><td>0</td><td>0</td><td>0</td><td>Card Data</td><td>MSRPAN</td><td>EXPAN</td><td>Amt</td></tr>
</table>

APPENDIX B. CREATING USER DATA

This is how each string of user data is created (used by Display Message and Request User Selection commands).

Bit	7	6	5	4	3	2	1	0
Byte 0	Length of parameters + string (no null at end of string) [includes this byte]							
Byte 1	X axis location of string (0-127)							
Byte 2	Y axis location of string (0-63)							
Byte 3	0 = background unchanged 1 = Background cleared	Set to 1 for underline	Spacing: 0 = Proportional 1 = Prop except #'s 2 = Fixed spacing		Alignment: 0 = Left 1 = Center 2 = Right		Font size: 0 = Small 1 = Small Bold 2 = Big	
Byte 4	0 (rfu)							
Bytes 5-n	String data (no terminating null)							

This is how the block of user data containing 1 or multiple user data strings as described above is created. This block of data must be sent using Report 0x10 – Send Big Block Data to Device before using this data in the Select or Display commands.

Bit	7	6	5	4	3	2	1	0
Byte 0	# of user data strings							
Byte 1..a1	User data string 1							
Byte a1+1..a2	Optionally User data string 2							
…	Etc…etc…etc							

```
void addUserString(MemoryStream m, byte x, byte y, byte p1, byte p2,string s)
{
    m.WriteByte((byte)(s.Length+5));
    m.WriteByte(x);
    m.WriteByte(y);
    m.WriteByte(p1);
    m.WriteByte(p2);

    System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();
    m.Write(encoding.GetBytes(s),0,s.Length);
}

~~~
MemoryStream ms = new MemoryStream();
ms.WriteByte(4);  // # of strings
addUserString(ms, 19, 56, 0x25, 0, "$20");
addUserString(ms, 64, 56, 0x25, 0, "$40");
addUserString(ms, 112, 56, 0x25, 0, "$100");
addUserString(ms, 64, 30, 0x15, 0, "Select Cashback");
pp.SendMultiData(6, ms.ToArray());  //6 for getsel, 7 for disp
pp.GetResponse(30, ResponseMsg.UserMsg, KeyMask.Left | KeyMask.Right |
KeyMask.Middle, 0);
~~~
```

APPENDIX C. GLOSSARY

API	Application Programming Interface
CRC	Cyclic Redundancy Check
DER	Distinguished Encoding Rules
DES	Data Encryption Standard. An algorithm developed in the1970s by the IBM Corporation, since adopted by the US government and ANSI (the American National Standards Institute) as the encryption standard for financial institutions.
DLL	Dynamically Linked Library
DUKPT	Derived Unique Key Per Transaction is a key management scheme in which a unique key is used for every transaction
EPB	Encrypted PIN Block
HID	Human Interface Device
KEY INJECTION	A secure operation whereby an encryption key is injected into a device
KSN	Key Serial Number
LCD	The Liquid Crystal Display is a 2-line by 16-character display that shows status, messages, and information on the magnetic stripe.
LED	The Light Emitting Diode is used for the power indicator on the dock.
MAGNEPRINT	MagnePrint is a card authentication technology which allows any magnetic stripe card to be recognized as a unique and non-reproducible security token. MagnePrint is able to detect cards that have been illegally reproduced (“skimmed”) as well as cards that have had their data re-encoded or magnetically altered. The term itself is derived from the following expressions: “Magne” as in magnetic and “Print” as in fingerprint.
MSR	Magnetic Stripe Reader
PAN	Personal Account Number
PIN	Personal Identification Number
PCI DSS	Payment Card Industry Data Security Standards
PCI PED	Payment Card Industry PIN Entry Device

PKI	Public Key Infrastructure (PKI) is an arrangement that binds public keys with respective user identities by means of a certificate authority.
rfu	Reserved for Future Use
TDES	Triple Data Encryption Standard
TRSM	Tamper-Resistant Security Module
USB	Universal Serial Bus