

# DynaPro Mini

**PIN Encryption Device**  
**Programmer's Manual (Commands)**



October 2018

Document Number:  
D99875629-43

REGISTERED TO ISO 9001:2015

Copyright © 2006 - 2018 MagTek, Inc.  
Printed in the United States of America

INFORMATION IN THIS PUBLICATION IS SUBJECT TO CHANGE WITHOUT NOTICE AND MAY CONTAIN TECHNICAL INACCURACIES OR GRAPHICAL DISCREPANCIES. CHANGES OR IMPROVEMENTS MADE TO THIS PRODUCT WILL BE UPDATED IN THE NEXT PUBLICATION RELEASE. 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.  
MagnePrint® is a registered trademark of MagTek, Inc.  
MagneSafe® is a registered trademark of MagTek, Inc.  
Magensa™ is a trademark of MagTek, Inc.  
DynaPro™ and DynaPro Mini™ are trademarks of MagTek, Inc.  
IPAD® is a registered trademark of MagTek, Inc.

AAMVA™ is a trademark of AAMVA.  
American Express® and EXPRESSPAY FROM AMERICAN EXPRESS® are registered trademarks of American Express Marketing & Development Corp.  
Apple Pay® is a registered trademark to Apple Inc.  
D-PAYMENT APPLICATION SPECIFICATION® is a registered trademark to Discover Financial Services CORPORATION  
MasterCard® is a registered trademark and PayPass™ and Tap & Go™ are trademarks of MasterCard International Incorporated.  
Visa® and Visa payWave® are registered trademarks of Visa International Service Association.

ANSI®, the ANSI logo, and numerous other identifiers containing "ANSI" are registered trademarks, service marks, and accreditation marks of the American National Standards Institute (ANSI).  
ISO® is a registered trademark of the International Organization for Standardization.  
PCI Security Standards Council® is a registered trademark of the PCI Security Standards Council, LLC.  
EMVCo™ and EMV™ are trademarks of EMVCo and its licensors.  
UL™ and the UL logo are trademarks of UL LLC.

Bluetooth® is a registered trademark of Bluetooth SIG.  
iPhone®, iPod®, and Mac® are registered trademarks of Apple Inc., registered in the U.S. and other countries. App Store™ is a service mark of Apple Inc., registered in the U.S. and other countries. iPad™ is a trademark of Apple, Inc. IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used by Apple Inc. under license.  
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.

All other system names and product names are the property of their respective owners.

**Table 0.1 - Revisions**

Rev Number	Date	Notes
1.01	Feb 18, 2014	Initial Release based on 99200078 7.01
2.01	Jul 25, 2014	Update based on 99200078-8.01: Update usage tables for 0x1A, 0xA1, 0xA2; apply consistent captions to tables and figures; add kernel IDs; clarify HID Usages; add 0x1F, 0x2E usages in report descriptor and elsewhere; add Bluetooth LE properties; misc. clarifications and accuracy fixes; add Report 0x06 bitmap option, Report 0x1F, Report 0x2E
22	Jun 22, 2015	Update a reference to an obsolete standard
30	Jul 17, 2015	Updated based on D9920007890: Clarify that bitmaps and contactless are supported only on DynaPro models; clarify Byte 47 of Command 0xA2; add control to disable checking inbound ARQC response's MAC; add Balance Read Before GenAC and Balance Read After GenAC in Command 0xA2; add Command 0x33 to read back whitelisted financial format BIN; add new signed configuration command to set financial format BIN for whitelisting; update trademark information; made financial+ICC card type reporting configurable; correct Merchant Data Container elements; clarify the number of slots in Command 0xA1; add example appendix illustrating how to parse EMV values from an SRED device; general cleanup and clarification
40	Feb 21, 2017	Misc. clarity improvements and fixes; address section / feature tagging issues; modify description of DFDF73 value 0 from unknown to default; revise device features table for DynaPro Mini firmware Rev D.; add back missing control for delayed response in Set feature of command 0xA1
41	Feb 27, 2017	Add transaction type 0x12 to <b>Command 0xA2 - Start EMV Transaction</b> ; correct usage of <b>Command 0x0E - Get Information</b>
42	Sep 7, 2018	From master programmer's manual Rev 150: Update <b>Table 1-1 - Device Features</b> ; Add information about Quick Chip functions and EMV flow to <b>Command 0xA2 - Start EMV Transaction</b> ; Update supporting section <b>About Message Authentication Codes ("MAC-AMK" or "MAC-MSR")</b> ; Misc. clarifications and corrections
43	Oct 18, 2018	Section <b>2.3</b> , replace device usage information with cross-reference to installation and operation manual; Throughout, replace BLE with Bluetooth LE.

## LIMITED WARRANTY

MagTek warrants that the products sold pursuant to this Agreement will perform in accordance with MagTek's published specifications. This warranty shall be provided only for a period of one year from the date of the shipment of the product from MagTek (the "Warranty Period"). This warranty shall apply only to the "Buyer" (the original purchaser, unless that entity resells the product as authorized by MagTek, in which event this warranty shall apply only to the first repurchaser).

During the Warranty Period, should this product fail to conform to MagTek's specifications, MagTek will, at its option, repair or replace this product at no additional charge except as set forth below. Repair parts and replacement products will be furnished on an exchange basis and will be either reconditioned or new. All replaced parts and products become the property of MagTek. This limited warranty does not include service to repair damage to the product resulting from accident, disaster, unreasonable use, misuse, abuse, negligence, or modification of the product not authorized by MagTek. MagTek reserves the right to examine the alleged defective goods to determine whether the warranty is applicable. Without limiting the generality of the foregoing, MagTek specifically disclaims any liability or warranty for goods resold in other than MagTek's original packages, and for goods modified, altered, or treated without authorization by MagTek.

Service may be obtained by delivering the product during the warranty period to MagTek (1710 Apollo Court, Seal Beach, CA 90740). If this product is delivered by mail or by an equivalent shipping carrier, the customer agrees to insure the product or assume the risk of loss or damage in transit, to prepay shipping charges to the warranty service location, and to use the original shipping container or equivalent. MagTek will return the product, prepaid, via a three (3) day shipping service. A Return Material Authorization ("RMA") number must accompany all returns. Buyers may obtain an RMA number by contacting MagTek Support Services at (888) 624-8350.

Each buyer understands that this MagTek product is offered as is. MagTek makes no other warranty, express or implied, and MagTek disclaims any warranty of any other kind, including any warranty of merchantability or fitness for a particular purpose.

If this product does not conform to MagTek's specifications, the sole remedy shall be repair or replacement as provided above. MagTek's liability, if any, shall in no event exceed the total amount paid to MagTek under this agreement. In no event will MagTek be liable to the buyer for any damages, including any lost profits, lost savings, or other incidental or consequential damages arising out of the use of, or inability to use, such product, even if MagTek has been advised of the possibility of such damages, or for any claim by any other party.

### LIMITATION ON LIABILITY

Except as provided in the sections relating to MagTek's Limited Warranty, MagTek's liability under this agreement is limited to the contract price of this product.

MagTek makes no other warranties with respect to the product, expressed or implied, except as may be stated in this agreement, and MagTek disclaims any implied warranty, including without limitation any implied warranty of merchantability or fitness for a particular purpose.

MagTek shall not be liable for contingent, incidental, or consequential damages to persons or property. MagTek further limits its liability of any kind with respect to the product, including any negligence on its part, to the contract price for the goods.

MagTek's sole liability and buyer's exclusive remedies are stated in this section and in the section relating to MagTek's Limited Warranty.

### **FCC WARNING STATEMENT**

This equipment has been tested and was found to comply with the limits for a Class B digital device pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a residential environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference with radio communications. However, there is no guarantee that interference will not occur in a particular installation.

### **FCC COMPLIANCE STATEMENT**

This device complies with Part 15 of the FCC Rules. Operation of this device is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

### **CANADIAN DOC STATEMENT**

This digital apparatus does not exceed the Class B limits for radio noise from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.


### **CE STANDARDS**

Testing for compliance with CE requirements was performed by an independent laboratory. The unit under test was found compliant with standards established for Class B devices.

### **UL/CSA**

This product is recognized per Underwriter Laboratories and Canadian Underwriter Laboratories 1950.

### **ROHS STATEMENT**

When ordered as RoHS compliant, this product meets the Electrical and Electronic Equipment (EEE) Reduction of Hazardous Substances (RoHS) European Directive 2002/95/EC. The marking is clearly recognizable, either as written words like "Pb-free," "lead-free," or as another clear symbol ()

## Table of Contents

LIMITED WARRANTY.....	4
FCC WARNING STATEMENT.....	5
FCC COMPLIANCE STATEMENT.....	5
CANADIAN DOC STATEMENT.....	5
CE STANDARDS.....	5
UL/CSA .....	5
RoHS STATEMENT.....	5
Table of Contents.....	6
<b>1 Introduction .....</b>	<b>10</b>
<b>1.1 About This Document .....</b>	<b>10</b>
<b>1.2 About Terminology .....</b>	<b>10</b>
<b>1.3 About Connection Types.....</b>	<b>10</b>
<b>1.4 About Device Features .....</b>	<b>10</b>
<b>1.5 About APIs.....</b>	<b>13</b>
<b>2 Connection Types.....</b>	<b>14</b>
<b>2.1 How to Use USB Connections (USB Only) .....</b>	<b>14</b>
<b>2.1.1 About HID Usages .....</b>	<b>14</b>
<b>2.1.1.1 About Reports .....</b>	<b>14</b>
<b>2.1.1.2 About the Report Descriptor .....</b>	<b>15</b>
<b>2.2 How to Use Apple iOS UART Connections (30-Pin Only).....</b>	<b>23</b>
<b>2.3 How to Use Bluetooth LE Connections (Bluetooth LE Only).....</b>	<b>24</b>
<b>3 Command Set .....</b>	<b>26</b>
<b>3.1 About Big Block Data and TLV Format.....</b>	<b>26</b>
<b>3.2 About SRED / Non-SRED Firmware (Non-SRED only).....</b>	<b>26</b>
<b>3.3 About Message Authentication Codes (“MAC-AMK” or “MAC-MSR”) .....</b>	<b>27</b>
<b>3.4 General Commands .....</b>	<b>30</b>
<b>3.4.1 Command 0x01 - Response ACK .....</b>	<b>30</b>
<b>3.4.2 Command 0x02 - End Session.....</b>	<b>30</b>
<b>3.4.3 Command 0x03 - Request Swipe Card .....</b>	<b>30</b>
<b>3.4.4 Command 0x04 - Request PIN Entry .....</b>	<b>32</b>
<b>3.4.5 Command 0x05 - Cancel Command.....</b>	<b>34</b>
<b>3.4.6 Command 0x06 - Request Cardholder Selection .....</b>	<b>34</b>
<b>3.4.7 Command 0x07 - Display Message .....</b>	<b>35</b>
<b>3.4.8 Command 0x08 - Request Device Status .....</b>	<b>36</b>
<b>3.4.9 Command 0x09 - Set / Get Device Configuration .....</b>	<b>36</b>
<b>3.4.10 Command 0x0A - Request MSR Data .....</b>	<b>40</b>

3.4.11	Command 0x0B - Get Challenge .....	41
3.4.12	Command 0x0D - Send Session Data - Amount .....	42
3.4.13	Command 0x0D - Send Session Data - PAN.....	42
3.4.14	Command 0x0E - Get Information .....	43
3.4.15	Command 0x0F - Login/Authenticate .....	45
3.4.16	Command 0x0F - Logout .....	46
3.4.17	Command 0x10 - Send Big Block Data to Device .....	46
3.4.18	Command 0x11 - Request Manual Card Entry.....	47
3.4.19	Command 0x14 - Request Cardholder Data Entry .....	49
3.4.20	Command 0x17 - Update Device.....	49
3.4.21	Command 0x1A - Request Device Information .....	50
3.4.22	Command 0x1C - Set/Get Bluetooth LE Power Configuration (Bluetooth LE Only) ....	54
3.4.23	Command 0x1D - Set Bluetooth LE Module Control Data (Bluetooth LE Only).....	54
3.4.24	Command 0x1E - Set iPod Accessory Protocol (iAP) Info (30-pin Only).....	55
3.4.25	Command 0x1E - Get iPod Accessory Protocol (iAP) Info (30-pin Only) .....	55
3.4.26	Command 0x30 - Set / Get KSN .....	55
3.4.27	Command 0x31 - Set KSN Encrypted Data .....	56
3.4.28	Command 0x32 - Get BIN Whitelist Table - Non Financial Format.....	58
3.4.29	Command 0x58 - Request Device Certificate .....	58
3.4.30	Command 0xFF - Device Reset.....	58
3.5	General Input Reports .....	60
3.5.1	Report 0x20 - Device State Report .....	60
3.5.2	Report 0x21 - Cardholder Data Entry Response Report .....	60
3.5.3	Report 0x22 - Card Status Report .....	61
3.5.4	Report 0x23 - Card Data Report.....	62
3.5.5	Report 0x24 - PIN Response Report.....	63
3.5.6	Report 0x25 - Cardholder Selection Response Report .....	64
3.5.7	Report 0x27 - Display Message Done Report .....	64
3.5.8	Report 0x29 - Send Big Block Data to Host.....	64
3.5.8.1	Big Block Data for Authorization Request (ARQC) .....	66
3.5.9	Report 0x2A - Delayed Response ACK .....	66
3.5.10	Report 0x2D - Bluetooth LE Module Control Data (Bluetooth LE Only) .....	67
3.6	EMV-Related Commands and Reports (EMV Only) .....	67
3.6.1	Report 0x2C - EMV Cardholder Interaction Status Report.....	67
3.6.2	Command 0xA1 - Access EMV Tags.....	69
3.6.2.1	Reading All EMV Tags .....	70
3.6.3	Command 0xA2 - Start EMV Transaction.....	71

3.6.3.1	Standard EMV Transaction .....	73
3.6.3.2	ARQC Request (EMV Only).....	75
3.6.3.3	Command 0xA2 Completion .....	78
3.6.4	Command 0xA4 - Acquirer Response ARPC (MAC-MSR) .....	79
3.6.5	Command 0xA8 - Get Kernel Info .....	79
3.6.6	Command 0xAB - Request EMV Transaction Data (MAC-MSR).....	81
3.6.7	Command 0xAC - Merchant Bypass PIN Command.....	87
Appendix A	Examples .....	88
A.1	How to Get MSR/PIN Data from the Device for a Bank Simulation .....	88
A.2	How to Parse Encrypted Big Block EMV Data From An SRED Device.....	91
Appendix B	Terminology .....	96
Appendix C	Status and Message Table .....	100
Appendix D	MagTek Custom EMV Tags (EMV Only).....	106
Appendix E	EMV Configurations (EMV Only) .....	111
Appendix F	Error Codes.....	112
F.1	H Codes .....	112
F.2	S Codes .....	113
F.3	C Codes .....	113
F.4	Device Offline K Codes .....	114
F.5	Device offline A Codes.....	114
Appendix G	Factory Defaults .....	115
G.1	Certificate Authority Public Keys .....	115
G.2	EMV Contact Factory Defaults (EMV Only).....	115
G.2.1	EMV Contact Terminal Factory Defaults.....	115
G.2.2	EMV Contact Payment Brand Factory Defaults.....	117
Appendix H	Language and Country Codes .....	118
H.1	Terminal Country Codes .....	118
H.2	Terminal Language Codes .....	118
Appendix I	Bluetooth LE Module Control Data (Bluetooth LE Only).....	119
I.1	Bluetooth LE Module Configuration Properties .....	119
I.1.1	Get Property Command .....	119
I.1.2	Set Property Command.....	119
I.1.3	Software ID Property .....	119
I.1.4	Bluetooth Device Address Property.....	120
I.1.5	Bluetooth Device Name Property .....	120
I.1.6	Configuration Revision Property .....	121
I.1.7	Power Timeout Property .....	121
I.1.8	Power Control Property .....	122



I.1.9	Advertising Control Property.....	123
I.1.10	Bluetooth LE Passkey Property.....	123
I.1.11	Desired Bluetooth LE Minimum Connection Interval Property .....	124
I.1.12	Desired Bluetooth LE Maximum Connection Interval Property .....	124
I.1.13	Desired Bluetooth LE Slave Latency Property .....	125
I.1.14	Desired Supervision Timeout Property .....	125
I.1.15	Connection Parameter Update Request Control Property .....	126
I.2	Other Commands .....	126
I.2.1	Echo Command .....	126
I.2.2	Reset Command .....	127
I.2.3	Erase All Non-volatile Memory Command .....	127
I.2.4	Erase All Bonds Command .....	127

# 1 Introduction

## 1.1 About This Document

This document describes the master command set available through byte-by-byte direct communication with DynaPro Mini PIN encryption devices (referred to in this document as “the device”).

## 1.2 About Terminology

The general terms “device” and “host” are used in different, often incompatible ways in a multitude of specifications and contexts. For example, “host” may have different a meaning in the context of USB communication than in the context of networked financial transaction processing. In this document, “device” and “host” are used strictly as follows:

- **Device** refers to the Pin Encryption Device (PED) that receives and responds to the command set specified in this document. Devices include DynaPro, DynaPro Mini, and so on.
- **Host** refers to the piece of general-purpose electronic equipment the device is connected or paired to, which can send data to and receive data from the device. Host types include PC and Mac computers/laptops, tablets, smartphones, teletype terminals, and even test harnesses. In many cases the host may have custom software installed on it that communicates with the device. When “host” must be used differently, it is qualified as something specific, such as “acquirer host” or “USB host.”

Similarly, the word “user” is used in different ways in different contexts. This document separates users into more descriptive categories:

- The **cardholder**
- The **operator** (such as a cashier, bank teller, customer service representative, or server), and
- The **developer** or the **administrator** (such as an integrator configuring the device for the first time).

Because some connection types, payment brands, and other vocabulary name spaces (notably Bluetooth LE, EMV, smart phones, and more recent versions of Windows) use very specific meanings for the term “Application,” this document favors the term **software** to refer to software on the host that provides a user interface for the operator.

The combination of device(s), host(s), software, firmware, configuration settings, physical mounting and environment, user experience, and documentation is referred to as the **solution**.

## 1.3 About Connection Types

DynaPro, DynaPro Mini, DynaPro Go, and related products use a common communication protocol across a variety of physical connection layers, which can include universal serial bus (USB), Ethernet, Apple 30-pin dock connector, and Bluetooth Low Energy (Bluetooth LE). The set of available connection layers depends on the device. Details for communicating with devices via each physical connection type are provided in section **2 Connection Types**.

## 1.4 About Device Features

The information in this document applies to multiple devices. When developing solutions that use a specific device or set of devices, integrators must be aware of each device’s communication interfaces, features, and configuration options, which affect the availability and behavior of some commands. **Table 1-1** provides a list of device features that may impact command availability and behavior.

Table 1-1 - Device Features

Feature	IPAD	DynaPro v1	DynaPro v3	DynaPro Plus	DynaPro Plus L1	DynaPro Mini 30-pin	DynaPro Mini Bluetooth LE	DynaPro Go	DynaPro Go PIN Function
General Features									
Signature Capture Stylus (“SC-S”)	Y	Y	Y	Opt	Opt	N	N	N	N
Signature Capture Finger (“SC-F”)	N	N	N	N	N	N	N	Y	Y
PIN Language Select	N	Y	Y	Y	Y	N	N	Y	Y
Re-PIN Support	N	N	N	N	N	N	N	N	Y
Custom Messages	Y	Y	Y	Y	Y	N	N	Y	Y
Bitmaps	Y	Y	Y	Y	Y	N	N	Y	Y
Clear Text User Data	N	Y	Y	Y	Y	N	N	Y	Y
Host-Supplied PAN	Y	Y	Y	Y	Y	Y	Y	N	N
Capacitive Keypad (“Cap Keypad”)	Y	Y	Y	Y	Y	N	N	N	N
Activation Codes	N	Y	Y	Y	Y	N	N	Y	Y
Fixed PIN Key	Y	Y	Y	Y	Y	Y	Y	N	N
PCI 4.x Key Block	N	N	N	N	N	N	N	Y	Y
IntelliHead	Y	Y	Y	Y	Y	Y	Y	N	N
Financial Format Whitelisting	N	Y	Y	Y	Y	N	N	Y	Y
Max Financial Card PAN Length	N	18	18	18	18	18	18	19	19
MagneSafe 2.0 (MS2.0)	Y	Y	Y	Y	Y	Y	Y	N	N
Token Reversal	N	Y	Y	Y	Y	Y	Y	N	N
Handheld Operation	N	N	N	N	N	N	N	Y	Y
Quick Chip	N	Y <sup>3</sup>	Y <sup>3</sup>	N	N	N	Y <sup>6</sup>	Y	Y
Beeper Control	N	Y	Y	Y	Y	Y	Y	Y	Y
Connections and Connection Features									
USB Connection	Y	Y	Y	Y	Y	Y	Y	Y	Y
TCP/IP Over 802.11 Wireless Connection	N	N	N	N	N	N	N	Y	N
Ethernet Connection	N	Opt	Opt	Opt	Opt	N	N	N	N
Static IP (Ethernet)	N	Y <sup>4</sup>	Y <sup>4</sup>	N	N	N	N	N	N
Apple 30-Pin Connection	N	N	N	N	N	Y	N	N	N

Feature	IPAD	DynaPro v1	DynaPro v3	DynaPro Plus	DynaPro Plus L1	DynaPro Mini 30-pin	DynaPro Mini Bluetooth LE	DynaPro Go	DynaPro Go PIN Function
RS-232 Connection	N	N	N	N	N	N	N	N	N
Bluetooth LE Connection	N	N	N	N	N	N	Y	N	N
<b>SRED Options</b>									
SRED	N	Opt	Opt	Opt	Opt	Opt	Opt	Y	Y
Non-SRED	Y	Opt	Opt	Opt	Opt	Opt	Opt	N	N
<b>EMV Features</b>									
Chip Card Contact	N	Y	Y	N	Y	Y	Y	Y	Y
Chip Card L1 Mode	N	N	N	N	Y	N	N	N	N
Chip Card L2 Mode	N	Y	Y	N	N	Y	Y	Y	Y <sup>2</sup>
RID CAPK Key Slots	N	8/16 <sup>1</sup>	8/16 <sup>1</sup>	N	N	8	8/16 <sup>1</sup>	16	16
Multiple Payment Brand Defaults	N	N	N	N	N	N	N	Y	Y
Chip Card Contactless	N	Opt	Opt	N	N	N	N	Y	Y <sup>2</sup>
MasterCard PayPass Support	N	Opt	Opt	N	N	N	N	N	N
MasterCard MCL 3.1.x support	N	N	N	N	N	N	N	Y	Y <sup>2</sup>
payWave 2.1.3 Support	N	Opt	Opt	N	N	N	N	N	N
payWave 2.2 Support	N	N	N	N	N	N	N	Y	Y <sup>2</sup>
Expresspay 3.0 Support	N	Opt	Opt	N	N	N	N	N	N
Expresspay 3.1 Support	N	N	N	N	N	N	N	Y	Y <sup>2</sup>
D-PAS Support	N	Opt	Opt	N	N	N	N	Y	Y <sup>2</sup>
Configurable EMV Support	N	Y	Y	Y	Y	N	N	N	N
<p>1) The number of CAPK key slots per RID depends on firmware revision number. DynaPro firmware up to revision D provides 8 key slots; revisions E and newer provide 16. DynaPro v3 provides 16 key slots. DynaPro Mini firmware up to revision C provides 8 key slots; revisions D and newer provide 16.</p> <p>2) Feature is not agency-certified.</p> <p>3) DynaPro v1 firmware revision F and newer; DynaPro v3 firmware revision B and newer.</p> <p>4) Available with Ethernet module firmware version 30050876-A00 and later.</p> <p>5) Available on DynaPro v3 with Contactless L1 – EMVCo Ver 2.6</p> <p>6) Available on DynaPro Mini Bluetooth LE firmware revision E or newer.</p>									

### 1.5 About APIs

MagTek provides convenient Application Programming Interface (API) libraries for some connection types and development frameworks. These APIs wrap the details of the connection in an interface that conceptually parallels the device's internal operation, freeing developers from dealing with the details of the connection, and allowing them to focus on software business logic. In cases where API libraries are available, developers also have the option to revert to direct communication with the device using libraries available in the chosen development framework. This document provides information and support for the latter method. Information about using MagTek APIs is available in separate documentation, including *D99875394 IPAD, DYNAPRO, AND DYNAPRO MINI PROGRAMMER'S MANUAL (.NET)*.

## 2 Connection Types

**Table 1-1** in section **1.4** includes a list of connection types available for each device. The following subsections provide details developers will need to communicate with the device using each connection type.

### 2.1 How to Use USB Connections (USB Only)

The device conforms to the USB specification revision 2.0, and are compatible with revision 1.1. It also conforms to the Human Interface Device (HID) class specification version 1.1, and communicates as a vendor-defined HID device. This document assumes the reader is familiar with USB HID class specifications, which are available at [www.usb.org](http://www.usb.org).

Developers can easily create custom software to communicate with the device using any framework that can make API calls to the standard Windows USB HID driver, such as Visual Basic or Visual C++. MagTek has developed demonstration software that communicates with the device via this method, and developers can use it to test the device and to provide a starting point for developing other software. For more information, see the MagTek web site, or contact your reseller or MagTek Support Services.

The device is a full speed high-powered USB device that, when connected, draws power from the USB bus. It identifies itself with vendor **ID 0x0801** and product ID **0x3009**. The device will enter and wake up from Suspend mode when directed to do so by the USB host. It does not support remote wakeup.

This device has programmable configuration properties stored in non-volatile memory. The properties are configured via the USB port and can be configured at the factory, by the key loader, or by the end user. More details can be found in section **3 Command Set** in this document, and in a separate document which provides details about key loading.

#### 2.1.1 About HID Usages

##### 2.1.1.1 About Reports

USB HID devices send and receive data using **reports**. Each report can contain several sections, called **usages**, each of which has its own unique four-byte identifier. The two most significant bytes of a usage are called the **usage page**, and the least two significant bytes are called the **usage ID**. Vendor-defined usages must have a usage page in the range **0xFF00 - 0xFFFF**, and it is common practice for related usage IDs to share the same usage page. For these reasons, all usages for this device uses vendor-defined usage page **0xFF20**.

HID reports used by the host can be divided into three types:

- **Feature Reports** (documented in section **3.4 General Commands**). Feature reports can be further divided into **Get** types and **Set** types. The host exclusively uses this type of report to send commands to the device and to receive synchronous responses from the device.
- **Input Reports** (documented in section **3.5 General Input Reports**) are used by the device to send asynchronous responses or notifications to the host when a related feature report completes, or automatically when the device's state changes. This is common when a command depends on cardholder action (for example, **Command 0x03 - Request Swipe Card** or **Command 0x04 - Request PIN Entry**) or otherwise takes more time to run.
- **Output Reports**. Output reports are part of the HID standard, but are not used by this device.

The host uses **HID Set** Feature Reports to send commands to the device, and **HID Get** Feature Reports to retrieve data or responses from the device when synchronous response is appropriate. The general sequence for using feature reports to send a command and receive a response is as follows:

- 1) Send the feature report (command), which could be either a Get or Set type.
- 2) Read **Command 0x01 - Response ACK** for acknowledgement, which includes the command number being acknowledged and a one-byte status indicating whether the device accepted the command.
- 3) For some commands, the host would then call a Get feature report to read the device's response.
- 4) For some commands, the host would instead expect the device to send an asynchronous response via an HID Input Report using a USB Interrupt IN transaction when the command finishes executing.

### 2.1.1.2 About the Report Descriptor

The list of the device's available reports and their structure is sent to the host in a **report descriptor**, usually just after the device is connected to the USB port. Generally the details of the report descriptor are abstracted by the developer's HID API; however, should it become necessary to examine a report descriptor byte-by-byte, a full inventory of the report descriptor for these devices is provided in **Table 2-1**, which also indicates whether each report is a Get type or Set type or both. The reports themselves are fully documented in the sections that follow.

**Table 2-1 - USB HID Report Descriptor**

Item	Value (Hex)
Usage Page	06 20 FF
Usage	09 01
Collection	A1 01
Report Size (8)	75 08
Logical Minimum (0)	15 00
Logical Maximum (255)	26 FF 00
Report ID (0x01) - Get	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 (0x02) - Set	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 (0x03) - Set	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 (0x04) - Set	85 04
Usage (Request PIN Entry)	09 04

## 2 - Connection Types

---

Item	Value (Hex)
Report Count (5)	95 05
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x05) - Set	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 (0x06) - Set	85 06
Usage (Request Cardholder Selection)	09 06
Report Count (4)	95 04
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x07) - Set	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 (0x08) - Set	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 (0x09) - Get/Set	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 (0x0A) - Set	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 (0x0B) - Get/Set	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 (0x0C) - Set	85 0C
Usage (Set Bitmap)	09 0C
Report Count (2)	95 02



## 2 - Connection Types

---

Item	Value (Hex)
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x0D) - Set	85 0D
Usage (Send Session Data/Send Session PAN)	09 0D
Report Count (21)	95 15
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x0E) - Get	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 (0x0F) - Set	85 0F
Usage (Authenticate/Logout)	09 0F
Report Count (9)	95 09
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x10) - Set	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
Report ID (0x11) - Set	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 (0x12) - Set	85 12
Usage (Request Cardholder Signature)	09 12
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x13) - Get	85 13
Usage (Get Cardholder Signature)	09 13
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x14) - Set	85 14
Usage (Request Cardholder Data Entry)	09 14
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01

## 2 - Connection Types

Item	Value (Hex)
Report ID (0x017) - Set	85 17
Usage (Update Device)	09 17
Report Count (8)	95 08
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x018) - Set	85 18
Usage (Perform Test)	09 18
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x019) - Get/Set	85 19
Usage (Extended Device)	09 19
Report Count (8)	95 08
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x01A) - Get/Set	85 1A
Usage (Request Device Configuration)	09 1A
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x1C) - Get/Set (Bluetooth LE only)	85 1C
Usage (Set/Get Bluetooth LE Power Configuration)	09 1C
Report Count (4)	95 04
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x1D) - Set (Bluetooth LE Only)	85 1D
Usage (Set Bluetooth LE Control Data)	09 1D
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x1E) - Get/Set (30-pin Only)	85 1E
Usage (Set/Get iPod Accessory Protocol [iAP] Protocol Info)	09 1E
Report Count (63)	95 3F
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0x1F)	
Usage (Request Clear Text Cardholder Data Entry)	85 1F
Report Count (3)	95 03
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0x20) - Input	85 20

## 2 - Connection Types

Item	Value (Hex)
Usage (Device State)	09 20
Report Count (6)	95 06
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x21) - Input	85 21
Usage (Cardholder Data Entry Response)	09 21
Report Count (20)	95 14
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x22) - Input	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 (0x23) - Input	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 (0x24) - Input	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 (0x25) - Input	85 25
Usage (Cardholder Selection Response)	09 25
Report Count (3)	95 03
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x27) - Input	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 (0x29) - Input	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
Report ID (0x2A) - Input	85 2A
Usage (Delayed Response ACK)	09 2A

## 2 - Connection Types

---

Item	Value (Hex)
Report Count (3)	95 03
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x2B) - Input	85 2B
RESERVED	
Report ID (0x2C) - Input	85 2C
Usage (EMV Cardholder Interaction Status)	09 2C
Report Count (127)	95 7F
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x2D) - Input (Bluetooth LE Only)	85 2D
Usage (Bluetooth LE Module Control Data)	09 2D
Report Count(64)	95 40
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x2E) - Input	85 2E
Usage (Clear Text Cardholder Data Entry Response Report)	09 2E
Report Count (12)	95 0C
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	82 02 01
Report ID (0x30 - Get/Set)	85 30
Usage (Set/Get KSN)	09 30
Report Count(1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0x31) - Set	85 31
Usage (Set KSN Encrypted Data)	09 31
Report Count(1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0x32) - Get/Set	85 32
Usage (Set/Get BIN Table)	09 32
Report Count(1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0x58) - Set	85 58
Usage (Key Handling or Manufacturing Command)	09 58
Report Count (2)	95 02
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA1) - Set	85 A1

## 2 - Connection Types

---

Item	Value (Hex)
Usage (Set or Get EMV Tag(s))	09 A1
Report Count (8)	95 08
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA2) - Set	85 A2
Usage (Request Start EMV Transaction)	09 A2
Report Count (48)	95 30
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA3) - Set	85 A3
Usage (Request ATR Data)	09 A3
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA4) - Set	85 A4
Usage (Acquirer Response)	09 A4
Report Count (12)	95 0C
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA5) - Set	85 A5
Usage (Set or Get CA Public Key)	09 A5
Report Count (8)	95 08
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA6) - Set	85 A6
Usage (Request Power Up/Down Reset ICC)	09 A6
Report Count (16)	95 10
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA7) - Get/Set	85 A7
Usage (Send/Get ICC APDU)	09 A7
Report Count (16)	95 10
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA8) - Set	85 A8
Usage (Get Kernel Info)	09 A8
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xA9) - Get/Set	85 A9
Usage (Get/Set Challenge and Session Key)	09 A9

## 2 - Connection Types

---

Item	Value (Hex)
Report Count (44)	95 2C
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xAA) - Set	85 AA
Usage (Confirm Session Key)	09 AA
Report Count (17)	95 11
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xAB) - Set	85 AB
Usage (Request EMV Transaction Data)	09 AB
Report Count (4)	95 04
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
Report ID (0xAC) - Set	85 AC
Usage (Merchant Bypass PIN Command)	09 AC
Report Count (1)	95 01
Feature (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Nvol,Buf)	B2 02 01
Report ID (0xFF) - Set	85 FF
Usage (Device Reset)	09 1E
Report Count (02)	95 02
Input (Data,Var,Abs,NWrp,Lin,Pref,Nnul,Buf)	B2 02 01
End Collection	C0

## 2.2 How to Use Apple iOS UART Connections (30-Pin Only)

When the device is connected to an iOS host via the **Apple 30-pin dock connector**, custom apps use iPod Accessory Protocol (iAP1) to communicate with the device using the `EASession` class. The custom software wraps commands in simple Get/Set wrappers, also called a UART packet header. The device firmware expects to receive and send data using the same formats produced by the `iAP iPodDataTransfer` and `AccessoryDataTransfer` commands, respectively. Documentation for these formats is available from Apple, specifically in *MFi Accessory Firmware Specification R44* (see <http://developer.apple.com/programs/mfi/>). Sample code is available in the form of Apple's **EADemo** app; see <https://developer.apple.com/library/IOS/samplecode/EADemo/Introduction/Intro.html>.

Because the device's command set is common to all connection types, it is also helpful to read section **2.1.1 About HID Usages** to become familiar with the types of available commands.

The devices only use **TXD** and **RXD**; hardware handshaking is not available. The serial settings are **57600 bps, No parity, 8 data bits, and 1 stop bit**. Code upgrade commands are not available through this connection. To communicate with a device using the UART connection, the host should begin all commands and responses with the following UART packet header:

**Table 2-2 - iOS UART Packet Header**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x00 = Get 0x01 = Set							
Byte 1..n	Command/Response as defined in section 3 <b>Command Set</b> .							

**IMPORTANT: Generally, iOS commands must be transmitted in MSB (big endian) order. By convention, this document gives commands in LSB (little endian) order.**

### 2.3 How to Use Bluetooth LE Connections (Bluetooth LE Only)

When the device is connected to a Bluetooth Low Energy (Bluetooth LE) capable host via Bluetooth LE, the device acts as a Bluetooth LE server/peripheral, and the host acts as a client/central. The host software wraps commands in simple Get/Set wrappers, and should use whatever Bluetooth LE library is appropriate for the chosen software development framework. For example, iOS host apps use Apple's CoreBluetooth Framework, for which sample code is available in the form of Apple's Temperature Sensor app (see <https://developer.apple.com/library/IOS/samplecode/TemperatureSensor/Introduction/Intro.html>).

Some of the details described in this section may be abstracted by the libraries in the chosen development framework. For general information about Bluetooth LE and the associated terms, see the Bluetooth specifications found at <https://www.bluetooth.org/Technical/Specifications/adopted.htm>.

The general steps for a host to communicate with the device via Bluetooth LE are as follows. Refer to **Table 2-3** for details about each Bluetooth LE Characteristic (in this case, "Application" refers to the device):

- 1) Make sure the device is advertising. Some devices do not need to be fully powered on to advertise. See the device's *Installation and Operation Manual* for specific steps.
- 2) Scan for nearby Bluetooth LE peripherals advertising the desired GATT service UUID.
- 3) If multiple devices of the desired type are available, examine each device's name property. A specific device's default name is a constant, equal to the product name plus a dash and a unique identifier (for example, part of the serial number, part of its Bluetooth address, etc.)
- 4) Establish a Bluetooth LE connection with the device. If this triggers the device to power on automatically, the device may take some number of seconds to fully power on. The host should not send additional commands or expect responses until after the power on sequence is complete.
- 5) Pair with the device. If the host software prompts for a passkey, see the device's *Installation and Operation Manual* for information about passkeys. In many cases this step is operator-driven.
- 6) Configure the device to notify the host when the Application Data To Host Length characteristic changes. The host should then use this notification as a trigger to read the Application Data To Host characteristic and process incoming data from the device. The specific method to enable notifications for a characteristic is different in different Bluetooth LE development libraries. For example, iOS code would be similar to `[servicePeripheral setNotifyValue:YES forCharacteristic:characteristic]`.
- 7) Send commands to the device by writing to the Application Data From Host Length characteristic, then to the Application Data From Host characteristic; receive notifications that the device has changed the Application Data To Host Length characteristic, and read the corresponding incoming data from the Application Data To Host characteristic.
- 8) For further information about device modes, power management, and usage, see the device's *Installation and Operation Manual*.

**Table 2-3 - Bluetooth LE Characteristics and UUIDs**

Characteristic	Max. Size	UUID In LSB (Little Endian) Order Some frameworks use MSB order
DynaPro Mini GATT service	N/A	01:01:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Application Data From Host Length	1 byte	20:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Application Data From Host	65 bytes	21:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05



## 2 - Connection Types

---

Characteristic	Max. Size	UUID In LSB (Little Endian) Order Some frameworks use MSB order
Application Data To Host Length	1 byte	22:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Application Data To Host	128 bytes	23:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05

To communicate with a device using the Bluetooth LE connection, the host should begin all commands and responses with the following header:

Bit	7	6	5	4	3	2	1	0
Byte 0	0x00 = Get 0x01 = Set							
Byte 1 ... Byte n	Request data as defined below and in section <b>3 Command Set</b> .							

Because the device's command set is common to all connection types, it would be helpful to read section **2.1.1 About HID Usages** to become familiar with the types of available commands before continuing.

The command / response for a Get report as described in section **2.1.1** are formatted as follows:

Request from Host:

Byte 0        0 (Get)  
Byte 1        Report ID

Response from Device:

Byte 0        Report ID  
Byte 1..n     Report  
Maximum report size is 63 bytes.

The command / response for a Set report as described in section **2.1.1** are formatted as follows:

Request from Host:

Byte 0        1 (Set)  
Byte 1        Report ID  
Byte 2..n     Report  
Maximum report size is 63 bytes.

Response from Device:

Byte 0        1 (Response ACK Report ID)  
Byte 1..2     (Response ACK Report)

Input report format:

Byte 0        Report ID  
Byte 1 - n     Report  
Maximum report size is 127 bytes.

### 3 Command Set

This section describes the full device command set. Because many commands are common across all connection types, it is helpful to first read and understand section **2.1.1 About HID Usages** to become familiar with the types of available commands.

#### 3.1 About Big Block Data and TLV Format

There are some cases where command data (host to device) or response data (device to host) requires special treatment. For example, some commands require the host or device to transmit large blocks of data that exceed the maximum packet size of the chosen data transport layer; other commands require transmitted data to be encrypted and/or encoded, fully received, then decrypted and/or decoded as a single piece. For commands and responses that require these sorts of special treatment, the usage information in this document indicates that the command or response uses **big block** data buffers.

The device provides support for transmitting big block data by implementing two reports: For feature reports that require big block data transmission, the host should first call **Command 0x10 - Send Big Block Data to Device** to transmit the relevant data to the device, then invoke the desired command. In cases where the device sends big block data to the host, the host will first invoke the desired command, then the device will send one or more instances of **Report 0x29 - Send Big Block Data to Host**. The host must then assemble / parse / decrypt / decode the data.

Big block data is frequently encoded using an industry standard Tag-Length-Value (TLV) format. For detailed information about parsing EMV response data in TLV format, see *EMV Integrated Circuit Card Specifications for Payment Systems 4.3, Part IV, Annex B Rules for BER-TLV Data Objects*. For a detailed example of parsing TLV data, see **Appendix A.2 How to Parse Encrypted Big Block EMV Data From An SRED Device**. For details about the specific tags used by a given command, see the usage information for the command.

#### 3.2 About SRED / Non-SRED Firmware (Non-SRED only)

The device can be loaded with one of two types of firmware, depending on how the integrator wants the device to transmit card data to the host:

- The SRED version of the firmware enables Secure Reading and Exchange of Data. In this mode, the device will not allow complete unmasking of card data, such as the PAN.
- In some cases, the solution may require further options for unmasking and encrypting card data before the device transmits it to the host. In those cases, the device can be loaded with the Non-SRED version of the firmware.

Some commands behave differently depending on whether the host is communicating with an SRED device or a non-SRED device. To determine which type of device the host is communicating with, use **Command 0x1A - Request Device Information** and examine the Capability String's SR parameter. The differences in command behavior between SRED and non-SRED devices are described in the usage information for each command.

### 3.3 About Message Authentication Codes (“MAC-AMK” or “MAC-MSR”)

“MAC” is an abbreviation of Message Authentication Code, which is a string of bytes included in a message that can be used to provide reasonable assurance that the message originated from a trusted source and has not been modified. All messages in this document (including commands, responses, and command payloads) that are tagged “MAC-AMK” or “MAC-MSR” must include the device’s unique serial number and a four-byte MAC.

The sections in this document about all commands, responses, and data formats that include a MAC are tagged with “MAC-AMK” or “MAC-MSR” in the section title. Some of these sections provide deep detail about generating and using the MAC, including which key and variant to use, which data elements to use, and how the resulting MAC is included in the message. The key used to calculate the MAC is usually either the MSR key or the AMK key, and the variant is always *Message Authentication, Request or Both Ways*. The choice of key depends on several factors, including the type of message, whether its related processes use encryption, and which encryption keys those processes use.

In all cases, the MAC is produced by following *ISO 9797-1 Information Technology – Security Techniques – Message Authentication Codes*, using Padding Method 1, Initial Transformation 1, Output Transformation 3, Algorithm 3, DEA, with two 56 bit-keys (K and K'). That method produces an 8-byte MAC value, and the most significant 32 bits of that value serve as the MAC the device or host will include with the message.

The host and device stage many MACed messages using big block data buffers (detailed in section **3.1 About Big Block Data and TLV Format**). In cases where the MACed message uses TLV data object F9, which is designed specifically for transmitting MACed messages, the message being sent as big block data follows this general format (interpret hexadecimal as binary values, ignore whitespace and /\*comments\*/, replace <angle bracketed values> with actual values):

```
AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
    DFDF54 (MAC KSN) <len><val>
    DFDF55 (MAC Encryption Type) <len><val>
    DFDF25 (IFD Serial Number) <len><val>
    <Nested TLV data objects specific to the message>
<Padding to force F9 plus padding to be a multiple of 8 bytes>
<Four byte CBC-MAC>
```

The host and device construct and send the full F9-based message as follows:

- The first two bytes of the message are in big-endian order (MSB first) not in TLV format, and indicate the length of the full F9 object, starting with the F9 byte at the beginning, and ending at the last byte of the F9 data object’s final nested child. By definition, this excludes any added padding and the CBC-MAC itself at the end of the message (both described later).
- The F9 data object is populated with nested TLV data objects as specified by the command, response, or data format’s documentation. In general:
  - If the MAC is generated using a DUKPT key, F9 will include nested TLV data object DFDF54 specifying the KSN for the DUKPT working key used to generate the MAC. If the MAC is generated using a fixed key, F9 will not include DFDF54.

- F9 will always include nested TLV data object DFDF55 containing a MAC Encryption Type, which specifies which key and variant was used to generate the MAC. See the definition of DFDF55 in **Appendix D MagTek Custom EMV Tags (EMV Only)** for information about valid values.
- F9 will always include nested TLV data object DFDF25 containing the device's unique IFD Serial Number, which the host can read from the device. See **Appendix D MagTek Custom EMV Tags (EMV Only)** for details about retrieving tags.
- The end of the message outside the F9 data object is padded to ensure that the length of data, starting with the F9 byte at the beginning, and ending with any additional padding, is a multiple of 8 bytes. This is a requirement of using the CBC-MAC algorithm.
- Using the key specified in the message's documentation, the CBC-MAC is calculated over the block of data that starts with the F9 byte at the beginning, through the last byte of any additional padding. This yields an 8-byte CBC-MAC.
- The final 4 bytes of the message are populated with the most significant 32 bits of the 8-byte CBC-MAC, not in TLV format.
- If the host is sending the message, it will send it to the device using **Command 0x10 - Send Big Block Data to Device**. If the device is sending the message as part of a response or report, it will send it to the host using **Report 0x29 - Send Big Block Data to Host**.



### 3.4 General Commands

#### 3.4.1 Command 0x01 - Response ACK

This command retrieves the status of the most recent command the host has sent to the device. The host should call this command immediately after it sends a command to the device, to determine whether the device accepted the command as sent. The data the device returns includes an ACKSTS code (see **Appendix C Status and Message Table**) and the ID of the command the status is for.

**Table 3-1 - Usage Table for Command 0x01**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x01							
Byte 1	Status of command (“ACKSTS”)							
Byte 2	ID of command being ACKed							
Byte 3	reserved							
Byte 4	reserved							

For example, after sending **Command 0x03 - Request Swipe Card** to the device, the host should immediately call **0x01 Response ACK**. If the command executed correctly, the ACKSTS would be 0x00; if the host included bad parameters in the command, the ACKSTS would be *Bad Parameter* (0x82), as specified in the documentation of **Command 0x03 - Request Swipe Card**.

#### 3.4.2 Command 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.



**Figure 3-1 - DynaPro Mini Welcome Screen**

**Table 3-2 - Usage Table for Command 0x02**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x02							
Byte 1	Idle Message ID: 0 = “Welcome” (default)							

#### 3.4.3 Command 0x03 - Request Swipe Card

This command directs the device to prompt the cardholder to swipe a card by displaying one of four predetermined messages (see Card Message ID in **Table 3-3**). Example request screens look like this:

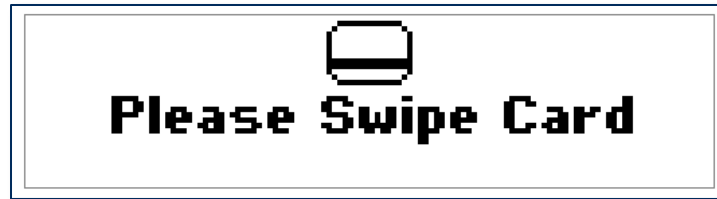


Figure 3-2 - DynaPro Mini Initial Swipe Prompt

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

When this command completes (card swiped OK, cardholder or operator cancelled, or timeout), the device will send **Report 0x22 - Card Status Report** to the host. If the Card and Operation Status are both OK, the host should then send **Command 0x0A - Request MSR Data** to get the card data.

For an example of using this command, see **Appendix A.1 How to Get MSR/PIN Data from the Device for a Bank Simulation**.

Table 3-3 - Usage Table for Command 0x03

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 4 = Chip Error, Use Mag Stripe							
Byte 3	Tones: 0 = No sound 1 = One beep 2 = Two beeps							

### 3.4.4 Command 0x04 - Request PIN Entry

This command directs the device to prompt the cardholder to enter a PIN by displaying one of five predetermined messages (see PIN Mode in **Table 3-4**).

The English on-screen prompts will look like this:



Figure 3-3 - DynaPro Mini Initial PIN Prompt

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, when the command completes (PIN entry done, cardholder or operator cancelled, or timeout), the device will send **Report 0x24 - PIN Response Report** to the host. 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 (see **Report 0x20 - Device State Report**). If there is no PAN (from card swipe or sent via command), the EPB will use ISO format 1. If a PAN exists, the PIN option will be used to determine if the created PIN block will be ISO format 0 or ISO format 3. If the host set the PIN Mode byte in the command to “Verify PIN,” the device will prompt the cardholder to enter the 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. If the host set the Wait Msg bit in the command’s PIN Options byte, the device will display a “Please Wait” message during the delay as the unit is checking for keypad tamper, then will display the Enter PIN message.

The selected language will remain active until the host sends **Command 0x02 - End Session**. The device will then switch to the default device language set in the contact database using **Command 0xA1 - Access EMV Tags**.

Table 3-4 - Usage Table for Command 0x04

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)			



### 3 - Command Set

---

Bit	7	6	5	4	3	2	1	0
Byte 4	Tones: 0 = No sound 1 = One beep 2 = Two beeps							
Byte 5	PIN options							
	RESERVED			RESERVED		Wait Msg	Verify PIN	ISO3

### 3.4.5 Command 0x05 - Cancel Command

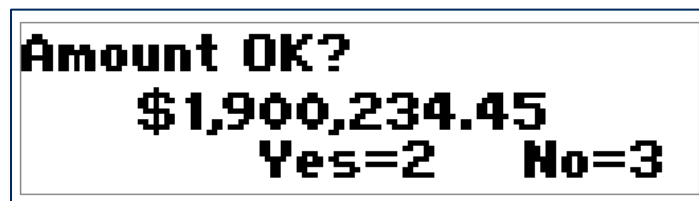
This command cancels the current command.

**Table 3-5 - Usage Table for Command 0x05**

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

### 3.4.6 Command 0x06 - Request Cardholder Selection

This command directs the device to prompt the cardholder to select the transaction type (debit, credit, etc.) or to verify the transaction amount, as shown below.



**Figure 3-4 - DynaPro Mini "Amount OK" Cardholder Selection Screen**

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

After the cardholder makes a selection or cancels, or if a timeout occurs based on Wait Time, the device will do the following:

- 1) Clear the display
- 2) Return to the idle state
- 3) Send **Report 0x25 - Cardholder Selection Response Report** to the host

**Table 3-6 - Usage Table for Command 0x06**

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 <b>Credit / Debit</b> 1 = Verify Transaction Amount 2 = Transaction type <b>Credit / Other / Debit</b> 3 = Transaction type <b>Credit / EBT / Debit</b> 4 = Transaction type <b>Credit / Gift/ Debit</b> 5 = Transaction type <b>EBT / Gift / Other</b>							
Byte 3	Mask Key:							
					Enter	Right	Middle	Left

Bit	7	6	5	4	3	2	1	0
Byte 4	Tone Pattern: 0 = No start beep, one timeout beep 1 = One start beep, one timeout beep 2 = Two start beeps, one timeout beep 100 = No start beep, no timeout beep 101 = One start beep, no timeout beep 102 = Two start beep, no timeout beep							

### 3.4.7 Command 0x07 - Display Message

This command directs the device to display a predefined message for a specified time. Examples are shown below.

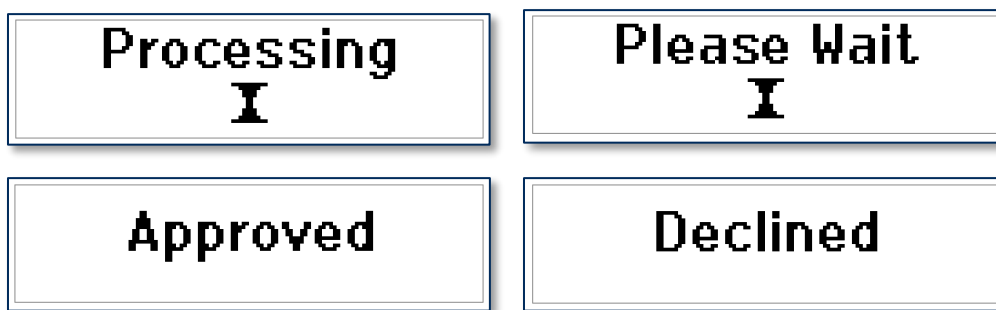


Figure 3-5 - DynaPro Mini Messages

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, when the command completes (message displayed, cardholder or operator cancelled, or timeout), the device will do the following:

- 1) Clear the display
- 2) Return to the idle state
- 3) Send **Report 0x27 - Display Message Done Report** to the host

Table 3-7 - Usage Table for Command 0x07

Bit	7	6	5	4	3	2	1	0
Byte 0	0x07							
Byte 1	Wait Time in seconds, (1 - 255; 0 = infinite wait time)							

Bit	7	6	5	4	3	2	1	0
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 10 = Call Your Bank 11 = CARD ERROR 12 = Not Accepted 13 = Processing Error 14 = Use CHIP READER 15 = Refer to your payment device							

### 3.4.8 Command 0x08 - Request Device Status

This command directs the device to send current information (Session State, Device State and Status, etc.) to the host. Following this command, the device will send the host **Report 0x20 - Device State Report**.

**Table 3-8 - Usage Table for Command 0x08**

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

### 3.4.9 Command 0x09 - Set / Get Device Configuration

The host calls this command in Set mode to send operator-defined configuration settings to the device. If the command changes the EMV Mode control setting, the host must send **Command 0xFF - Device Reset** or the operator must power cycle the device before that setting will take effect.

The host can also call this command in Get mode to direct the device to return its current configuration settings. The format of the device's response in this case is identical to the format of the host's request when calling the command in Set mode (**Table 3-9**).

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will save the new configuration.

The Configuration Lock bit can only be set by the manufacturer. When the device is locked, the host cannot change any of the device configuration settings. When the device is unlocked, the host can only change the following settings:

### 3 - Command Set

---

- MSR Encryption Variant
- Clear Text Cardholder Data
- Beeper Mode
- Mask Configuration
- MSR Card Configuration
- Mask Character
- Number of leading/trailing to leave unmasked
- EMV L2 ICS Configuration
- Financial+ICC Card Type reporting

All other settings, including the Configuration Lock bit, can only be changed by the supplier or manufacturer.

**Table 3-9 - Usage Table for Command 0x09 (Set mode) / Response for Command 0x09 (Get mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x09							
Byte 1	Device Control (default value = 0x00)							
	Bit 0 Require Authentication: 0 = No 1 = Yes  Bit 1 MSR Encryption Variant: 0 = PIN 1 = DATA  Bit 2 Reserved  Bit 3 Reserved  Bit 4 iPod Accessory Protocol (iAP) Config Allowed: (30-pin Only) Allows configuration of Reverse DNS and Bundleseed ID when unlocked 0 = Unlocked 1 = Locked  Bit 4 Reserved  Bit 5 Reserved  Bit 6 Reserved  Bit 7 Configuration Lock (NOTE: After locking, unlocking can only be performed by the manufacturer): 0 = Unlocked 1 = Locked							

Bit	7	6	5	4	3	2	1	0
Byte 2	Device Control 2 (default value = 0x00)							
	Bit 0 Allow Charging in iPod Accessory Protocol (iAP) mode (30-pin Only) Configure charging mode 0 = Disable (default) 1 = Enable							
	Bit 1 MAC checking control for inbound online response (EMV Only) Configure MAC requirement on inbound online response for <b>Command 0xA4 - Acquirer Response ARPC (MAC-MSR)</b> : 0 = Enable (default) 1 = Disable							
	Bit 2 Financial + ICC Card Type Reporting: Configure Financial+ICC Card type reporting 0 = Enable (default) 1 = Disable							
Byte 3	Mask Configuration (default value = 0xC0, all enabled except MS2.0)							
	ISO Mask 0 = Disable 1 = Enable	Check Digit 0 = Disable 1 = Enable	MS2.0 Enable 00 = MS2.0 Disabled		MS2.0 Only: Track 2 Data		MS2.0 Only: Track 1 Data	
			MS2.0 Only: 10 = MS2.0 Enabled		Error	Blank	Error	Blank
	MSR Card Configuration (default value = 0xD5, all enabled)							
Byte 4	AAMVA Card 0 = Disable 1 = Enable	Non-finance card option	Track 3 Data 00 = Disabled 01 = Enabled 11 = Required		Track 2 Data 00 = Disabled 01 = Enabled 11 = Required		Track 1 Data 00 = Disabled 01 = Enabled 11 = Required	
			Mask Character (factory setting is 0x30 representing ASCII '0')					
Byte 6	Leading length to leave unmasked In SRED, maximum length = 6				Trailing length to leave unmasked			
	MS2.0 Only: When the device is configured to use MagneSafe MS2.0 format, values greater than 8 are interpreted as 8; values less than 5 are interpreted as 5				In SRED, maximum length = 4 Ignored in MS2.0 format			
Byte 7	EMV L2 ICS Configuration (Default = 0x01) Note: This setting is ignored when EMV Mode is disabled.							

Bit	7	6	5	4	3	2	1	0
					EMV L2 ICS Configuration 0000 = No L2 capability 0001 = Configuration C1 0010 = Configuration C2  DynaPro Mini firmware Rev D and newer: 0011 = Configuration C3 0100 = Configuration C4 0101 = Configuration C5 0110 = Configuration C6 0111 = Configuration C7 1000..1111 = Reserved  DynaPro Mini Rev A to C: Configurations C1 and C2 are EMVCo certified.  DynaPro Mini Rev D and newer: Configurations C1, C4 and C5 are EMVCo certified.			
Byte 8	Default = 0x00  Bits 0 through 7 Reserved							

Notes for Byte 3, bits 0..3 (MS2.0 Only):

- If Error = 0, the device will build MS2.0 format track data (if MS2.0 is enabled) if at least one track contains good data. The indicated track may contain errors.
- If Error = 1, the device will not build MS2.0 format track data (if MS2.0 is enabled) if the indicated track contains error(s).
- If Blank = 0, the device will build MS2.0 format track data (if MS2.0 is enabled) if at least one track contains good data. The indicated track may be blank.
- If Blank = 1, the device will not build MS2.0 format Track data (if MS2.0 is enabled) if the indicated track is blank.

### 3.4.10 Command 0x0A - Request MSR Data

This command directs the device to send MSR data to the host; it should be issued after a **Command 0x03 - Request Swipe Card** command has successfully completed.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will send multiple instances of **Report 0x23 - Card Data Report** to the host. If no MSR data is available, the device will send a single **Report 0x23 - Card Data Report** containing a Data Length of 0.



**Table 3-10 - Usage Table for Command 0x0A**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0A							
Byte 1	0x00							

### 3.4.11 Command 0x0B - Get Challenge

This command directs the device to send challenge information to the host. The host should first issue the command in Set mode as follows:

**Table 3-11 - Usage Table for Command 0x0B (Set mode)**

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 0x0A = Inject Configuration signed by PIN cert 0x0B = Inject Configuration signed by MSR cert 0x20...0x29 = RESERVED 0x63 = Authentication 0xFF = MFG command *Note: Use MSR Key when getting challenge to inject Acquirer Master Key							

After sending this command to the device and getting the ACKSTS report, the host should issue this command in Get mode. If the key ID is not in the list, or a valid authentication key is not available for key ID = 0x63, the data block will be all zeros.

**Table 3-12 - Usage Table for Command 0x0B (Get mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0B							

Bit	7	6	5	4	3	2	1	0
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 0x0A = Inject Configuration signed by PIN cert 0x0B = Inject Configuration signed by MSR cert 0x20...0x29 = RESERVED 0x63 = Login/Logout/Authentication 0xFF = MFG command *Note: Use MSR Key when getting challenge to inject Acquirer Master Key							
Byte 2..13	Data block: If Key ID < 12 or Key ID = 0xFF: Bytes 2..9 contain the device serial number Bytes 10..13 contain the random token  If Key ID = 0x63 and a valid authentication key is available: Bytes 2..9 contain the encrypted partial device serial number and random token Bytes 10..13 contain the partial device serial number							

### 3.4.12 Command 0x0D - Send Session Data - Amount

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

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

**Table 3-13 - Usage Table for Command 0x0D (For Amount)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0D							
Byte 1	0x00							
Byte 2	String length of transaction amount: 1-11							
Byte 3	Reserved for future use							
Byte 4..14	Amount data in ASCII format							
Byte 15..21	Reserved							

### 3.4.13 Command 0x0D - Send Session Data - PAN

This command is used to send card PAN data to the device in cases where the PAN is coming from a source other than the card being processed.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

**Table 3-14 - Usage Table for Command 0x0D (For PAN)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0D							
Byte 1	0x01							
Byte 2	String length of PAN data: 8-19							
Byte 3..21	PAN data in ASCII format							

#### 3.4.14 Command 0x0E - Get Information

The host uses this command to retrieve various types of information from the device.

**Table 3-15 - Usage Table for Command 0x0E**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0E							
Byte 1	Info ID (see <b>Table 3-17</b> )							

If the host is using any Info Id other than 0x80, it should follow these steps:

- 1) Call this command in Set mode, specifying the Info ID from **Table 3-17**. If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.
- 2) Call this command in Get mode with the same parameters.
- 3) The device responds with the requested data.

If the host is retrieving Info ID 0x80, the host should simply call this command in Get mode, and if the device has keys injected, the command will return the KCV/Hash value of the last injected key or certificate.

**Table 3-16 - Usage Table for Command 0x0E**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0E							
Byte 1	Info ID (see <b>Table 3-17</b> )							
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 <b>Table 3-17</b> )							
Byte 3	Data length (see <b>Table 3-17</b> ); default value is 0							

Bit	7	6	5	4	3	2	1	0
Byte 4..63	Block data							

Table 3-17 - Response for Command 0x0E

Info ID	Key Status	Data length	Data	Description
0x00	1	lblen*	Auth key label	If authorization 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 authorization key exists
0x06	1	<=19	Label and KCV	If fixed key exists
0x08	1	12	Customer ID (4 byte integer), Signing Sequence Number (4 byte integer), Upgradability Options (4 byte integer)	0x00000000 is Generic Customer Signing Sequence typically starts at 0x00000000 and must advance with each upgrade.  Upgradability values: 0x00000000 = Generic Only 0x00000001 = Specific Only 0x00000002 = Generic or Specific
0x09	1	<=19	Label and KCV	Acquirer Master Key
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
0x20 - 0x29	-	-	-	RESERVED
0x50	1	9	Keypad sensitivity Tamper sensitivity Key on threshold Key off threshold 4 bytes keypad threshold Keypad calibration result	Keypad values (Cap Keypad Only)
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***

Info ID	Key Status	Data length	Data	Description
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	Device authorization key signed by PIN cert
0x80	kcv_type=5	4	Hash value	Device authorization key signed by MSR cert
0x80	kcv_type=9	4	KCV value	KCV for Acquirer Master key
0x80	All other kcv_types	0		KCV****
* lblen = authorization 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				

### 3.4.15 Command 0x0F - Login/Authenticate

The host uses this command to authenticate with the device (log in) or to revoke authentication (log out).

The host must follow these steps to initiate authentication:

- 1) Request an authentication token from the device using **Command 0x0B - Get Challenge**
- 2) Decrypt the received token with the authentication key
- 3) Transform the token and encrypt it with the authentication key
- 4) Call the Login / Authenticate form of this command.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, authentication is successful.

**Table 3-18 - Usage Table for Command 0x0F (For Login/Authenticate)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0F							
Byte 1	0x00 = Logout 0x01 = Login / Authenticate							

Bit	7	6	5	4	3	2	1	0
Byte 2..9	If logging in, encrypted random token and device serial number (8 bytes). See <b>Command 0x0B - Get Challenge</b> . If logging out, Reserved.							

### 3.4.16 Command 0x0F - Logout

This command logs out the device.

**Table 3-19 - Usage Table for Command 0x0F (For Logout)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x0F							
Byte 1								
Byte 2..9	reserved							

### 3.4.17 Command 0x10 - Send Big Block Data to Device

This command is used to provide data for several general commands and input reports (listed in **Table 3-20**) in 60-byte increments. If the data size is greater than 60 bytes, the data must be split into several smaller blocks, each containing a maximum of 60 bytes. Two data formats are used in connection with this command: The first packet (block 0, see **Table 3-20**) signals the start of a new data set and specifies the complete length of the data; subsequent packets (blocks 1 through n, see **Table 3-21**) transmit the actual data to a predefined buffer within the device. For more information about big blocks and TLV format, see section **3.1 About Big Block Data and TLV Format**.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the bitmap image will be stored in a predefined buffer within the device for use by a subsequent command.

If the Data Type byte is set to one of the types tagged “Secured” or “MAC,” the first two bytes of the Packet Data in Block 1 must begin with a two-byte header in big-endian form (MSB first) that contains the expected total length of the data for the command (all blocks 1..n), excluding data padding and CBC-MAC.

**Table 3-20 - Usage Table for Command 0x10 (Block 0)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x10							
Byte 1	Data Type: 0x17 = Firmware file 0xA0 = EMV data in DOL format (MAC) 0xA1 = EMV data in TLV format, Set EMV Tag(s) (MAC) 0xA4 = EMV data in TLV format, Acquirer Response (ARPC) 0xA5 = CA Public Key Data (MAC)							

Bit	7	6	5	4	3	2	1	0
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 (if Byte 7 indicates Legacy) Data Length middle low byte (if Byte 7 indicates Extended)							
Byte 5	Reserved (if Byte 7 indicates Legacy) Data Length middle high byte (if Byte 7 indicates Extended)							
Byte 6	Reserved (if Byte 7 indicates Legacy) Data Length high byte (if Byte 7 indicates Extended)							
Byte 7	Extended Type: 0 = Legacy, data length less than 64K 1 = Extended, data length > 64K							
Byte 8..63	Reserved							

**Table 3-21 - Usage Table for Command 0x10 (Blocks 1 through n)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x10							
Byte 1	Data type: 0x17 = Firmware file 0xA1 = EMV data in TLV format 0xA4 = EMV data in TLV format, Acquirer Response (ARPC) 0xA5 = CA Public Key Data (MAC)							
Byte 2	Data packet number (1...n)							
Byte 3	Packet length							
Byte 4..63	Packet data: For EMV data, use Tag-Length-Value format. For more information about big blocks and TLV format, see section <b>3.1 About Big Block Data and TLV Format</b> .							

### 3.4.18 Command 0x11 - Request Manual Card Entry

This command directs the device to prompt the cardholder to enter the following card information by keypad. Two modes are available: Account Number mode and Qwick Code mode:

**Account Number mode** prompts for the following:

- Account number (minimum length = 9, maximum length = 19)
- Expiration date (minimum length = maximum length = 4)
- Card verification code (minimum length = 3, maximum length = 4)

**Qwick Code mode** prompts for the following:

- Qwick Code (minimum length = 8, maximum length = 16)
- Last 4 digits of account # (minimum length = maximum length = 4)
- Card verification code (minimum length = 3, maximum length = 4)

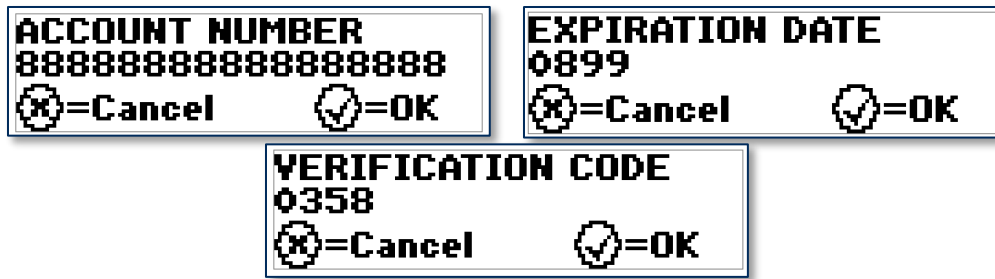


Figure 3-6 - DynaPro Mini Manual Card Information Entry

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will send **Report 0x22 - Card Status Report** to the host. If the cardholder or operator cancelled the request, or if the request timed out, byte 1 of **Report 0x22 - Card Status Report** will contain the appropriate Operation Status code to indicate why the command did not complete. Otherwise, if all of the card information was entered correctly, 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 **Command 0x0A - Request MSR Data**). The device will respond with multiple instances of **Report 0x23 - Card Data Report**, followed by **Report 0x20 - Device State Report**.

Table 3-22 - Usage Table for Command 0x11

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	0	0	0=Use PAN min 9, max 19 1=Use PAN min 14, max 21	1=Use PAN in PIN block creation	1=Use 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							

The track data sent by the device for manually entered card data may be masked according to the device’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’ indicates the field options used when the data was collected; unused fields will be 0s. 0s 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 string literal “MANUAL ENTRY?”.

Track 1 data may be found in the instance of **Report 0x23 - Card Data Report** that contains Data ID = 0x01. The device will format Track 1 card data as follows:



```
%M5555555555555555^MANUAL ENTRY/^33330000004444000006?
```

Track 2 data may be found in the instance of **Report 0x23 - Card Data Report** that contains Data ID = 0x02. The device will format Track 2 card data as follows:

```
;5555555555555555=33330000004444006?
```

The device does not change the length of the CVC (either 3 or 4 characters) entered by the cardholder. The length of the CVC thus affects the length of the track data output by the device, 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 (?).

### 3.4.19 Command 0x14 - Request Cardholder Data Entry

This command directs the device to prompt the cardholder to enter SSN, Zip Code, Birth Date, or Activation Code (firmware revision C12 and newer) by displaying one of four predetermined messages.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

Otherwise, when the command completes (data entry done, cardholder or operator cancelled, or timeout), the device will send **Report 0x21 - Cardholder Data Entry Response Report** to the host. 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.

**Table 3-23 - Usage Table for Command 0x14**

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	Cardholder 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							

### 3.4.20 Command 0x17 - Update Device

This command directs the device to validate, authenticate, and use the file data to upgrade the device's main application firmware. It is only available when using the USB or Ethernet connections. Further

information about the bootloader is available in document *99200071 MNL DYNALEX BOOTLOADER*.

To send new file data to the device, follow these steps:

- 1) Issue **Command 0x10 - Send Big Block Data to Device** to send new file data to the device
- 2) Issue **Command 0x17 - Update Device** to request the device to validate and perform an update using the file data with the correct parameters.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

**Table 3-24 - Usage Table for Command 0x17**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x17							
Byte 1	Options. See <b>Table 3-25</b> .							
Byte 2..8	Reserved							

**Table 3-25 - Options for Command 0x17**

Byte 1 Usage									
	7	6	5	4	3	2	1	0	Notes
Reset Options	N/A	N/A	N/A	N/A	N/A	N/A	0	0	Return Response ACK (big data must exist), Update, Reset when Done
	N/A	N/A	N/A	N/A	N/A	N/A	0	1	Return Response ACK (big data must exist), Update, Delayed Response ACK, Allow Reset later
	N/A	N/A	N/A	N/A	N/A	N/A	1	0	RFU
	N/A	N/A	N/A	N/A	N/A	N/A	1	1	Return Response ACK, wait 100ms, Reset Now (only valid after successful update)
RFU	X	X	X	X	X	X	N/A	N/A	RFU

### 3.4.21 Command 0x1A - Request Device Information

This command requests information about the device. Use this command in Set mode first to specify the information to retrieve, then use Get mode to retrieve the requested information.

**Table 3-26 - Usage Table for Command 0x1A**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	Mode							
	0 = Product_ID 1 = Maximum Device Message Size 2 = Capability String 3 = Manufacturer 4 = Product Name 5 = Serial Number 6 = Firmware Number 7 = Build Info A = Boot1 Firmware Version B = Boot2 Firmware Version 0x10 = Contactless Database Status (for future release)							
Byte 2..63	Reserved							

Depending on what device information the host has requested, the device will respond to this command using the following formats.

**Table 3-27 - Usage Table for Command 0x1A - Product\_ID**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x00							
Byte 2..63	"3004" (null terminated string)							

**Table 3-28 - Usage Table for Command 0x1A - Maximum Device Message Size**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x01							
Byte 2 to 63	"64" (null terminated string)							

**Table 3-29 - Usage Table for Command 0x1A - Capability String**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x02							

### 3 - Command Set

Bit	7	6	5	4	3	2	1	0
Byte 2..63	“V=4,SC=1,SR=1,TR=1,MS2=1,PFK=1,UDE=2,CE=2,CLE=1,DR=1” (null terminated string)  Capability String Code Description: V = Device Type SC = Signature Capture Support, 1=Supported, 0=Not Supported SR = SRED, 1=SRED, 0=NON-SRED TR = Token Reversal Support, 1=Supported, 0=Not Supported MS2 = MagneSafe 2.0 Support, 1=Supported, 0=Not Supported PFK = PIN Fixed Key Support, 1=Supported, 0=Not Supported UDE = Cardholder Data Entry Mode, 1=Encrypted Only, 2=Clear Text and Encrypted CE = Contact EMV Level Support, 1=L1, 2=L2 CLE = Contactless EMV Level Support, 1=L1, 2=L2 DR = Delayed Response Support, 1=Supported, 0=Not Supported							

**Table 3-30 - Usage Table for Command 0x1A - Manufacturer**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x03							
Byte 2..63	“MagTek, Inc.” (null terminated string)							

**Table 3-31 - Usage Table for Command 0x1A - Product Name**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x04							
Byte 2..63	The product name is a null-terminated string containing “DynaPro Mini”							

**Table 3-32 - Usage Table for Command 0x1A - Serial Number**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x05							
Byte 2..63	“12345678” (null terminated string)							

**Table 3-33 - Usage Table for Command 0x1A - Firmware Number**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x06							

### 3 - Command Set

---

Bit	7	6	5	4	3	2	1	0
Byte 2..63	"30050856A01-DEMO" (null terminated string)							

**Table 3-34 - Usage Table for Command 0x1A - Build Info**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x07							
Byte 2..63	<date><time>" (null terminated string)							

**Table 3-35 - Usage Table for Command 0x1A - Boot1 Firmware Version**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x0A							
Byte 2..63	"30050858B01-DEMO" (null terminated string)							

**Table 3-36 - Usage Table for Command 0x1A - Boot2 Firmware Version**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1A							
Byte 1	0x0B							
Byte 2..63	"30050866B02-DEMO" (null terminated string)							

### 3.4.22 Command 0x1C - Set/Get Bluetooth LE Power Configuration (Bluetooth LE Only)

This command sets or gets the device’s Bluetooth LE power configuration, depending on whether it is called in Set mode or Get mode.

**Table 3-37 - Usage Table for Command 0x1C**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1C (command identifier)							
Byte 1	Reserved, set to 0	Reserved, set to 0	Reserved, set to 0	Reserved, set to 0	Reserved, set to 0	Ready	Suspend	Power down
Byte 2	Reserved, set to 0x00							
Byte 3	Reserved, set to 0x00							
Byte 4	Reserved, set to 0x00							

Power Down bit: (Read/Write)

0: The device is not powered down.

1: The device is powered down.

Powering down the device will conserve power. The host will have limited communications capability with the device when it is powered down. During this time, the device will only respond to command 0x1C. Powering the device down will conserve more power than suspending it. It takes more time to power the device up than it does to take it out of suspend.

Suspend bit: (Read/Write)

0: The device is not suspended.

1: The device is suspended.

Suspending the device will conserve power. The host will have limited communications capability with the device when it is suspended. During this time, the device will only respond to command 0x1C. Powering the device down will conserve more power than suspending it. It takes more time to power the device up than it does to take it out of suspend.

Ready bit: (Read only, always write 0)

0: The device is not ready.

1: The device is ready.

The host will have limited communications capability with the device when it is powered down or suspended. During this time, the device will only respond to command 0x1C. Because it takes some time for the device to be capable of full communication after powering it up or taking it out of suspend, this flag can be read to determine whether the device is ready for full communication.

### 3.4.23 Command 0x1D - Set Bluetooth LE Module Control Data (Bluetooth LE Only)

This command sends control data to the Bluetooth LE module, which controls Bluetooth LE communications. The device will first respond with **Command 0x01 - Response ACK**. After the Bluetooth LE module control data is processed, the device will respond with **Report 0x2D - Bluetooth LE Module Control Data (Bluetooth LE Only)**. It is important to understand that even though this is implemented as a feature report to be used in Set mode, it can be used to send requests to discover the current property settings in the Bluetooth LE module. The full set of Bluetooth LE module controls can be found in **Appendix I Bluetooth LE Module Control Data**.

**Table 3-38 - Usage Table for Command 0x1D**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1D (command identifier)							
Byte 1	Control data length (defined in <b>Appendix I</b> )							
Byte 2 to (2+control data length-1)	Control data (defined in <b>Appendix I</b> )							
Byte (2 + control data length) to 63	Padding. Set all bytes to zero.							

### 3.4.24 Command 0x1E - Set iPod Accessory Protocol (iAP) Info (30-pin Only)

When the host calls this command in Set mode, it sets iPod Application Protocol (iAP) related data in the device.

**Table 3-39 - Usage Table for Command 0x1E (Set mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1E (command identifier)							
Byte 1	0 = Set Bundle Seed ID (10 bytes) 1 = Set reverse DNS (variable, up to 50 bytes)							
Byte 2	Data Length							
Byte 3..63	Data							

### 3.4.25 Command 0x1E - Get iPod Accessory Protocol (iAP) Info (30-pin Only)

When the host calls this command in Get mode, it retrieves the current device iPod Application Protocol (iAP) protocol Bundle Seed ID and reverse DNS in the following format:

**Table 3-40 - Usage Table for Command 0x1E (Get mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x1E (command identifier)							
Byte 1	Length of Bundle Seed ID, normally 0x0A (<=0x0A)							
Byte 2..11	Bundle Seed ID, high bytes padded with 0x00 if length is less than 0x0A							
Byte 12	Reverse DNS Length (<= 0x32)							
Byte 13..63	Reverse DNS, high bytes padded with 0x00 if length is less than 0x32							

### 3.4.26 Command 0x30 - Set / Get KSN

When called in Set mode, this command directs the device to generate a KSN data for transmission to a host.

**Table 3-41 - Usage Table for Command 0x30 (Set mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x030							
Byte 1	Key ID: 0x00 = MSR DUKPT key KSN							

After sending this command to the device and getting the ACKSTS report, issue the same command in Get mode for the KSN Feature Report (see **Table 3-42**). If a valid DUKPT key is not available, the data block will be all zeros.

The KSN reported is only valid for 1 minute. **Command 0x31 - Set KSN Encrypted Data** should be sent within the timeout period.

This feature is used for the Token Reversal Function.

**Table 3-42 - Usage Table for Command 0x30 (Get mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x30							
Byte 1	Key ID: 0x00 = MSR DUKPT key KSN							
Byte 2..11	Data block: Bytes 2..11 contain the KSN							
Byte 12..19	Device Serial Number							
Byte 20..23	Padding							
Byte 24..27	CBC-MAC							

### 3.4.27 Command 0x31 - Set KSN Encrypted Data

Before using this command, the host must have already used **Command 0x30 - Set / Get KSN** to retrieve the MSR DUKPT KSN from the device. Then the host must use **Command 0x10 - Send Big Block Data to Device** to send encrypted PAN data to the device, in the following format:

```
AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
  DFDF54 (MAC KSN)<len><val>
  DFDF55 (MAC Encryption Type)<len><val>
  DFDF25 (IFD Serial Number)<len><val>
FA<len> /* container for generic data */
  DFDF44 (Encrypted PAN data)<len><val>
  (Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, use MAC variant of MSR dukpt key)
```

The host can then use this command to direct the device to process data in the big block. The device decrypts and displays the data until the display timeout expires.

This feature is used for the Token Reversal Function.



The value of DFDF44 is always encrypted under the data variant of the MSR DUKPT key.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

**Table 3-43 - Usage Table for Command 0x31**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x31							
Byte 1	Display Time in seconds, (1 - 255; 0 = 256 seconds)							

### 3.4.28 Command 0x32 - Get BIN Whitelist Table - Non Financial Format

This command will cause the device to send the current contents of the BIN whitelist for non-financial cards to the host in the following format:

**Table 3-44 - Usage Table for Command 0x32 (Get mode)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x32							
Byte 1	Number of bytes to follow (36 for the 6 slots)							
Byte 2..7	Data on BIN Table Slot 1							
Byte 8..13	Data on BIN Table Slot 2							
Byte 14..19	Data on BIN Table Slot 3							
Byte 20..25	Data on BIN Table Slot 4							
Byte 26..31	Data on BIN Table Slot 5							
Byte 32..37	Data on BIN Table Slot 6							

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

### 3.4.29 Command 0x58 - Request Device Certificate

This command directs the device to send the device certificate to the host.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will send **Report 0x29 - Send Big Block Data to Host** to the host.

**Table 3-45 - Usage Table for Command 0x58 (For Device Cert)**

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

### 3.4.30 Command 0xFF - Device Reset

This command directs the device to perform a restart.

**Table 3-46 - Usage Table for Command 0xFF**

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

### 3 - Command Set

---

Bit	7	6	5	4	3	2	1	0
Byte 1	0 = Soft Reset							
Byte 2	Reserved							

### 3.5 General Input Reports

Input reports are asynchronous data packets (i.e., events) sent from the device to the host. Events occur when the device state changes, or upon completion of an asynchronous command (such as a command that requires cardholder input) sent by the host to the device.

#### 3.5.1 Report 0x20 - Device State Report

This event is triggered explicitly when the host successfully issues **Command 0x08 - Request Device Status**, or automatically when the device changes state. Both cases cause the device to send Device State, Session State, Device Status, Device Certificate Status, and Hardware Status to the host.

**Table 3-47 - Usage Table for Report 0x20**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x20							
Byte 1	Device State (see <b>Appendix C Status and Message Table</b> )							
Byte 2	Session State (see <b>Appendix C Status and Message Table</b> )							
Byte 3	Device Status (see <b>Appendix C Status and Message Table</b> )							
Byte 4	Device Certificate Status (see <b>Appendix C Status and Message Table</b> )							
Byte 5	Hardware Status (see <b>Appendix C Status and Message Table</b> )							
Byte 6	ICC Master and Session Key Status Bit 0: 1 = No Acquirer Master Key Injected  Bit 1 1 = No ICC Session Key Active  Bit 2: 1 = CAPK EMV database corrupted  Bit 3: 1 = EMV Terminal / Payment Brand Database corrupted  Bit 4: 1 = Card Present in chip card connector  Bit 5: RESERVED							

#### 3.5.2 Report 0x21 - Cardholder Data Entry Response Report

This event supports **Command 0x14 - Request Cardholder Data Entry**. After the cardholder has successfully entered data, the device uses this report to send cardholder data to the host.

**Table 3-48 - Usage Table for Report 0x21**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x21							

Bit	7	6	5	4	3	2	1	0
Byte 1	Operation Status (see <b>Appendix C Status and Message Table</b> )							
Bytes 2..11	MSR KSN							
Bytes 12..19	Encrypted Cardholder Data block (ECDB)							

The ECDB contains the information that was requested by the host with **Command 0x14 - Request Cardholder Data Entry** (for example, if the host requested the cardholder’s zip code, this report would return just the zip code data). After decryption, The 8-byte Cardholder Data Block is divided into 16 four-bit nybbles, as specified in the tables below. Each nybble contains one of the following:

- C: Control field
  - 0100=SSN
  - 0101=Zip Code
  - 0110=Birth Date
- N: Data length
- P: Cardholder data digit from 0000 (decimal 0) to 1001 (decimal 9)
- R: Filled random number
- P/R: If the Birth Date data length is 6 (MMDDYY format), the positions marked P/R will be filled with random numbers (R); if the Birth Date data length is 8 (MMDDYYYY format), those positions will contain the rightmost two characters of the birth year (P).

**Table 3-49 - Report 0x21 Cardholder Data Block Format**

Bits	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31	32-35	36-39	40-43	44-47	48-51	52-55	56-59	60-63
SSN	C	N	P	P	P	P	P	P	P	P	P	R	R	R	R	R
Zip code	C	N	P	P	P	P	P	R	R	R	R	R	R	R	R	R
Birth date	C	N	P	P	P	P	P	P	P/R	P/R	R	R	R	R	R	R
Activation Code	C	N	P	P	P	P	R	R	R	R	R	R	R	R	R	R

### 3.5.3 Report 0x22 - Card Status Report

This event is triggered by **Command 0x03 - Request Swipe Card** or **Command 0xA2 - Start EMV Transaction**, which will cause the device to send Operation Status, Card Status, and Card Type to the host.

**Table 3-50 - Usage Table for Report 0x22**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x22							
Byte 1	Operation Status (see <b>Appendix C Status and Message Table</b> )							
Byte 2	Card Status (see <b>Appendix C Status and Message Table</b> )							

Bit	7	6	5	4	3	2	1	0
Byte 3	Card Type (see <b>Appendix C Status and Message Table</b> )							

### 3.5.4 Report 0x23 - Card Data Report

In response to the host sending **Command 0x0A - Request MSR Data** after a card swipe or manual card entry, the device will send one instance of this report for each Data ID listed in **Table 3-51**.

**Table 3-51 - Usage Table for Report 0x23**

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 0x40 = Encrypted PAN and expiration date (financial cards only; otherwise data is blank) 0x41 = Device serial number 0x63 = KSN and MagnePrint Status 0x64 = CBC-MAC							
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, Bytes 4 -13 are KSN data; bytes 14-17 are MP Status data If Data ID = 0x41, data is 8 byte serial number If Data ID = 0x64, data is 4 byte CBC-MAC If Data ID = 0x40, data is encrypted PAN and Expiration date in the following format: Start Sentinel(‘;’) PAN Separator (‘=’) YYMM (‘?’)							

MS2.0 Only: If the device has been configured to use the MS2.0 masking configuration (see **Command 0x09 - Set / Get Device Configuration**), then track status (byte 2) of Data ID 0x63 uses a different set of status values, defined as follows:

**Table 3-52 - Report 0x23 Track Status Byte When Using MS2.0 Masking (MS2.0 Only)**

Value	Track Status If Using MS2.0
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

### 3.5.5 Report 0x24 - PIN Response Report

This event is triggered by **Command 0x04 - Request PIN Entry**, which directs the device to send PIN data to the host after a PIN is successfully entered.

The device may report ‘Keypad Security’ in Byte 1 to indicate the keypad has detected a tamper condition. This can be triggered by a cardholder pressing a function key for too long when selecting an account type. To cover this case, the software should include retry logic that resends **Command 0x04 - Request PIN Entry** to re-arm the PIN pad.

**Table 3-53 - Usage Table for Report 0x24**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x24							

Bit	7	6	5	4	3	2	1	0
Byte 1	Operation Status (see <b>Appendix C Status and Message Table</b> )							
Bytes 2..11	PIN KSN. If fixed PIN key is used, KSN is zero.							
Bytes 12..19	Encrypted PIN block							

### 3.5.6 Report 0x25 - Cardholder Selection Response Report

The device will send this report to the host to provide a response to **Command 0x06 - Request Cardholder Selection**.

**Table 3-54 - Usage Table for Report 0x25**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x25							
Byte 1	Operation Status (see <b>Appendix C Status and Message Table</b> )							
Byte 2	Code of key pressed: 0x71 = left function key 0x72 = middle function key 0x74 = right function key 0x78 = ENTER key							

### 3.5.7 Report 0x27 - Display Message Done Report

The device sends this report to the host to indicate a pending **Command 0x07 - Display Message** has completed.

**Table 3-55 - Usage Table for Report 0x27**

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

### 3.5.8 Report 0x29 - Send Big Block Data to Host

The device sends this report to the host to send the cardholder’s signature, device certificate, or CSR to the host. If the data size is greater than 123 bytes, the data must be broken into multiple 123-byte data blocks, or packets. The host will use three distinct types of block 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.

If the big block buffer type parameter is one of the types tagged with “Secured” or “MAC,” the first two bytes of the Data Block in Block 1 are the expected total length of the response data (from all blocks 1..n), excluding data padding and CBC-MAC.



If the big block buffer type parameter is PayPass Asynchronous Message, after assembly of the different packets, the first two bytes of the data packet are the length, and the TLV formatted message starts on the third byte.

For more information about big blocks and TLV format, see section **3.1 About Big Block Data and TLV Format**.

**Table 3-56 - Start of Big Block Sending Format (Block 0)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x29							
Byte 1	Big block buffer type: 0x02 = Device certificate 0x32 = Set BIN (MAC) 0x42 = CSR 0xA1 = EMV data in TLV format, Tag Data (MAC) 0xA2 = RESERVED 0xA3 = RESERVED 0xA4 = EMV data in TLV format, Authorization Request (ARQC) 0xA5 = CA Public Key (MAC) 0xAB = EMV data in TLV format, Batch Data or Batch Data and Reversal Data							
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							

**Table 3-57 - Big Block Data Sending 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..n	Data block (maximum 123 bytes)							

**Table 3-58 - End of Big Block Sending Format**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x29							
Byte 1	Not defined							
Byte 2	99 = End flag							

### 3.5.8.1 Big Block Data for Authorization Request (ARQC)

In response to **Command 0xA2 - Start EMV Transaction**, when an ONLINE approval is required as determined by the device and the ICC, by default the device will send EMV tag 70 containing an authorization request message to the host in the format shown below. The list of tags the device will include in the authorization request message can be re-configured by changing the list of tags stored in tag DFDF02 (see **Command 0xA1 - Access EMV Tags**). The data will be contained in EMV tag 70.

**Table 3-59 - Default ARQC Data Format**

Tag	Description	Source	Format	Length
9F03	Secondary amount associated with the transaction representing a cash back amount	Device	n	6
9F26	Cryptogram returned by the ICC in response of the GENERATE AC command	Card	b	8
82	EMV Application Interchange Profile	Card	b	2
5A	EMV Application PAN	Card	c	0-10
5F34	EMV Application PAN Sequence Number	Card	n	1
9F36	EMV Application Transaction Counter	Card	b	2
9F1A	EMV Terminal Country Code	Device	n	2
95	TVR	Device	b	5
9F02	Authorized amount of the transactions (excluding adjustments)	Device	n	6
5F2A	Transaction Currency Code	Device	n	2
9A	Local Date transaction was authorized	Device	n	3
9C	Transaction Type	Device	n	1
9F37	Unpredictable Number	Device	b	4
9F10	Issuer EMV Application Data	Card	b	0-32
DFDF53	Fallback Indicator	Device	n	1
F5	Container For Encrypted PIN	Device	b	Var
F4	Container For Encrypted MSR	Device	b	Var

### 3.5.9 Report 0x2A - Delayed Response ACK

This event is triggered by completion of longer-running commands that need to report status back to the host. It is an asynchronous version of **Command 0x01 - Response ACK**.

**Table 3-60 - Usage Table for Report 0x2A**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x2A							
Byte 1	Status of Command (“ACKSTS”)							
Byte 2	ID for Command reporting status							

Bit	7	6	5	4	3	2	1	0
Byte 3..n	Reserved							

### 3.5.10 Report 0x2D - Bluetooth LE Module Control Data (Bluetooth LE Only)

This input report event is triggered by completion of **Command 0x1D - Set Bluetooth LE Module Control Data (Bluetooth LE Only)** after the device returns **Command 0x01 - Response ACK** and processes the control data. Important usage notes regarding getting and setting Bluetooth LE module properties are included in section **3.4.23 Command 0x1D - Set Bluetooth LE Module Control Data (Bluetooth LE Only)**.

**Table 3-61 - Usage Table for Report 0x2D**

Byte	Description
Byte 0	0x2D
Byte 1	Control data length (defined in <b>Appendix I</b> )
Bytes 2 to (2 + control data length - 1)	Control data (defined in <b>Appendix I</b> )
Bytes (2 + control data length) to 63	Padding. All bytes are zeros.

## 3.6 EMV-Related Commands and Reports (EMV Only)

This section contains both commands sent from the host to the device (feature reports) and asynchronous events sent from device to the host (input reports) that support EMV transaction processing.

After the device successfully reads a chip card, it generates EMV data in the form of **tags** for transaction processing. The device then sends the host its own information plus information read from the card. The host will generally then use that information to authorize, complete, and save a transaction.

If fallback is enabled, the device will use magnetic stripe data to process a transaction if it can not read the chip card.

A number of tags can be configured on the device using the Set form of **Command 0xA1 - Access EMV Tags**, such as EMV terminal floor limit, terminal ID, and transaction currency code.

### 3.6.1 Report 0x2C - EMV Cardholder Interaction Status Report

This event is triggered during an EMV transaction started by **Command 0xA2 - Start EMV Transaction**. Events are generated when there is a cardholder interaction; for example, when a screen is displayed and waits for cardholder input. This report is used to update the merchant display throughout the transaction based on cardholder interactions.

**Table 3-62 - Usage Table for Report 0x2C**

Bit	7	6	5	4	3	2	1	0
Byte 0	0x2C							

### 3 - Command Set

Bit	7	6	5	4	3	2	1	0
Byte 1	EMV Cardholder Interaction Status ID: 0x01 = Waiting for amount confirmation selection 0x02 = Amount confirmation selected 0x03 = Waiting for multi-payment ICC Application selection 0x04 = ICC Application selected 0x07 = Waiting for language selection 0x08 = Language selected 0x09 = Waiting for credit/debit selection 0x0A = Credit/Debit selected 0x0B = Waiting for Pin Entry for ICC 0x0C = PIN entered for ICC 0x0D = Waiting for Pin Entry for MSR 0x0E = PIN entered for MSR							
Byte 2	0x00 (RESERVED)							
Byte 3	0x00 (RESERVED)							
Byte 4..127	Data block: If EMV Cardholder Interaction Status ID from Byte 1 = 0x02, value 0x1 indicates Amount Confirmed, or value 0x2 indicates Amount Not Confirmed.  If EMV Cardholder Interaction Status ID from Byte 1 = 0x04, data is a string representing EMV application preferred name, or label chosen by cardholder.  If EMV Cardholder Interaction Status ID from Byte 1 = 0x0A, value 0x1 indicates Credit, or value 0x2 indicates Debit.  If EMV Cardholder Interaction Status ID from Byte 1 = 0x20, bytes 4 and 5 will indicate the length of data starting at byte 6, which is in TLV format.  Otherwise, no data.							

### 3.6.2 Command 0xA1 - Access EMV Tags

This command gives the host access to slots on the device which store one or more EMV tags that drive the device's EMV kernel configurations. The set of available configuration tags combines multiple sources:

- The EMV specification, *EMV Integrated Circuit Card Specifications for Payment Systems v4.3*.
- The MagTek custom tags listed in **Appendix D MagTek Custom EMV Tags** (for example, the host can assign the database of EMV tags a label using tag DFDF26, and read the checksum back using tag DFDF27);

### 3.6.2.1 Reading All EMV Tags

To read all EMV tags stored on the device, the host should send the following command:

**Table 3-63 - Usage Table for Command 0xA1 (Set form to Read All Tags)**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xA1							
Byte 1	Specifies which EMV or Contactless tag group to read:							
	00 = EMV/Contactless Terminal tag group			If bits 6 and 7 are set to EMV/Contactless Application, bits 0-5 specify which application slot to read.				
	10 = EMV/Contactless Application tag group			Number of supported slots: Contact: 0..9 (Contact Chip Card Only)				
Byte 2	0x0F=Read all EMV terminal or payment brand tags							
Byte 3	Database Selector: 00 = Contact L2 EMV Tags							
Byte 4..8	Reserved							

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will send **Command 0x01 - Response ACK**, then send **Report 0x29 - Send Big Block Data to Host** with the EMV tags and requested data.

### 3.6.3 Command 0xA2 - Start EMV Transaction

This command directs the device to initiate various transactions for magnetic stripe cards, chip cards, and contactless cards. The host should send the command to the device as follows:

**Table 3-64 - Usage Table for Command 0xA2**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xA2							
Byte 1	Wait time in seconds, (0x01..0x3C) for cardholder to confirm, cancel, and present card. This timer is also used for the cardholder to choose an ICC application.							
Byte 2	Wait time in seconds, (0x01..0x3C) for cardholder to enter PIN.							
Byte 3	0x00, Reserved							
Byte 4	Beeper Behavior 0 = No sound 1 = One beep 2 = Two beeps							
Byte 5	Card Type to Read: 0x01 = Magnetic Stripe Card 0x02 = Contact Chip Card  Multiple Card Types can be selected by ORing the values together. For example: Set byte 5 to 0x03 to read both Magnetic stripe card and contact chip card.							
Byte 6	Options: 0x00 = Normal 0x01 = Bypass PIN 0x02 = Force Online 0x04 = Acquirer not available (Note: prevents long timeout on waiting for host approval)							
Bytes 7..12	Amount Authorized (EMV Tag 9F02, n12 format)							
Byte 13	Transaction Type:  DynaPro (Firmware Rev A to D) and DynaPro Mini (Firmware Rev A to C) 0x02 or 0x09 = Cashback 0x04 = Goods (Purchase) 0x08 = Services (Purchase)  DynaPro (Firmware Rev E and newer) and DynaPro Go and DynaPro Mini (Firmware Rev D and newer) 0x00 = Purchase 0x01 = Cash Advance 0x02 or 0x09 = Cashback 0x04 = Goods (Purchase) 0x08 = Services (Purchase) 0x12 = Cash Manual 0x50 = Payment (Chip Card Contact Only)							
Bytes 14..19	Cashback Amount (if non-zero, EMV Tag 9F03, n12 format)							
Bytes 20..25	Reserved							

### 3 - Command Set

---

Bit	7	6	5	4	3	2	1	0
Bytes 26..31	Reserved							
Bytes 32..33	Transaction Currency Code (EMV Tag 5F2A, format n4) Valid values: 0x0000 = Use value from contact database 0x0840 = US Dollar 0x0978 = Euro 0x0826 = UK Pound							
Byte 34	Reserved							
Byte 35	0x00 = Regular Mode 0x01 = Quick Chip Mode							
Byte 36..63	Reserved. Set = 0x00							

If an error occurs, the device terminates the command and reports the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Command 0x01 - Response ACK**.



#### 3.6.3.1 Standard EMV Transaction

The sequence for calling the command is as follows:

- 1) As a cautionary measure, the host may send the device **Command 0x02 - End Session** to make sure the device does not have a previous unfinished transaction lingering, which would block initiation of a new transaction.
- 2) The host finalizes the amount for the transaction.
- 3) The host sends the device **Command 0xA2 - Start EMV Transaction**.
- 4) The device prompts the cardholder to present payment by displaying pre-defined EMV messages:
  - a) DynaPro Mini cycles between showing **(AMOUNT)**, **(AMOUNT) OK?** and **CANCEL OR ENTER**, and waits for the cardholder to select **Enter** or **Cancel**.
- 5) If the cardholder confirms the amount or makes a payment type selection, then depending on the card type the host specified in the command, the device arms the MSR and/or ICC slot, turns on the LED near the slot (if any), and displays either **SWIPE** or **INSERT CARD**. If the cardholder presses the **Cancel** button or the transaction times out, the device performs **Command 0xA2 Completion**.
- 6) If the cardholder swipes a magnetic stripe card, the device meets EMV 4.x requirements by checking the service code from the magnetic stripe data to see if it begins with a 2 or a 6 to determine if the card also includes a chip, and advises the cardholder that EMV is preferred by displaying **USE CHIP READER** (similar to the screenshots for **Command 0x07 - Display Message**). If the chip fails or the service code does not begin with a 2 or a 6, the device prompts the cardholder for an MSR swipe. After a successful swipe, the device prompts the cardholder to select **Debit** or **Credit**. If this is a debit account type, the device requests a PIN.
- 7) If the cardholder inserts a contact chip card, depending on the device's payment brand account type setting for ICC the Acquirer has set in tag DFDF73 [see **Appendix D MagTek Custom EMV Tags (EMV Only)**] the device does one of the following:
  - a) Assume Credit, Debit, or Default.
  - b) Prompt the cardholder to select **Credit**, **Default**, or **Debit**.
- 8) If the cardholder has inserted a contact chip card, the device shows ICC applications that are mutually supported by both the card and the device, and asks the cardholder to choose the preferred ICC application. If there is no mutually supported application, the device may show **CARD BLOCKED**. If a PIN entry is needed per **EMV 4.x** requirements, the device shows **ENTER PIN** and starts the PIN entry timer (similar to the screenshots for **Command 0x04 - Request PIN Entry**). If the cardholder cancels the transaction or the transaction times out, the device performs **Command 0xA2 Completion**. When the device is configured to allow PIN bypass using tag DFDF68, the PIN requirement can be bypassed by the merchant by setting bit 0, byte 6 of the 0xA2 command. The TVR bits will be set appropriately per EMV 4.x requirements. The PIN requirement can also be bypassed by the cardholder.
- 9) After PIN entry, the device displays either **PIN OK** or cycles through **INCORRECT PIN** and **TRY AGAIN** up to the PIN retry limit (similar to the screenshots in **Command 0x04 - Request PIN Entry**). If the number of attempts reaches PIN try limit-1, the device displays **Last PIN Try** briefly before returning to **TRY AGAIN**. If the cardholder exceeds the PIN entry retry limit, the device performs **Command 0xA2 Completion**, otherwise the transaction proceeds to the approval stage.
- 10) The device determines the appropriate transaction approval method per **EMV 4.x** requirements. A transaction can be forced online by the merchant by setting the **Options** in the invocation of Command 0xA2 to **Force Online**.
  - a) For offline transactions, the device gets the TC or AAC from the chip for later transmission to the host. Depending on the transaction outcome, the device shows **APPROVED**, **DECLINED**, or

**ERROR** (similar to the screenshots in **Command 0x07 - Display Message**) and performs **Command 0xA2 Completion**.

- b) For online transactions, the device does the following:
- i) If the host started the transaction with Quick Chip Mode in effect:
    - (1) The device sends an **ARQC Request (EMV Only)** to the host for approval using **Report 0x29 - Send Big Block Data to Host**. The host should save this information for later verification.
    - (2) The device immediately constructs its own ARPC Response with tag 8A set to 'Z3' and continues the transaction processing.
    - (3) The device performs **Command 0xA2 Completion**.
    - (4) The device shows **REMOVE CARD** to notify cardholder the card can be removed, and ends the transaction.
    - (5) The host should then process the ARQC Message data, including setting the final transaction amount, and should coordinate with the transaction processor to retrieve a final transaction result.
    - (6) Based on the transaction result, the host can show **APPROVED** or **DECLINED** using **Command 0x07 - Display Message**.
    - (7) The host may request for the transaction data by sending **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**.
  - ii) If the host started the transaction with Regular Mode (Quick Chip Mode NOT in effect):
    - (1) The device sends an **ARQC Request (EMV Only)** to the host for approval using **Report 0x29 - Send Big Block Data to Host**.
    - (2) The device starts a host response timer.
    - (3) The device waits for the host to send **Command 0xA4 - Acquirer Response ARPC (MAC-MSR)**.
    - (4) The device processes the host response.
    - (5) The device gets TC or AAC from the chip.
    - (6) Depending on the transaction outcome, the device displays **APPROVED**, **DECLINED** or **ERROR** (similar to the screenshots in **Command 0x07 - Display Message**), and performs **Command 0xA2 Completion**.

### 3.6.3.2 ARQC Request (EMV Only)

ARQC stands for Authorization Request Cryptogram. These are requests sent by the payment method to the device to the host, which then transmits them to the transaction processor to request approval for the transaction.

#### 3.6.3.2.1 Non-SRED ARQC Request (Non-SRED Only, MAC-MSR)

On non-SRED devices, the device sends ARQC messages using **Report 0x29 - Send Big Block Data to Host** with a message structured like the example below. The host may also customize the contents of the messages using tag DFDF02.

For details about using the general F9 MAC structure, see section **3.3 About Message Authentication Codes (“MAC-AMK” or “MAC-MSR”)**. Information specific to this message is provided after the example.

```

AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */

    DFDF54 (MAC KSN)<len><val>
    DFDF55 (MAC Encryption Type)<len><val>
    DFDF25 (IFD Serial Number)<len><val>
    FA<len> /* container for generic data */
        70<len> /*container for ARQC */
            DFDF53<len><value> /*fallback indicator */
            5F20<len><value> /*cardholder name */
            5F30<len><value> /*service code */
            DFDF4D<len><value> /* Mask T2 ICC Data */
            DFDF52<len><value> /* card type */
            F4<len> /* container tag for encrypted MSR data, if
present */
                DFDF36 <EncT1status><len><val>
                DFDF37 <EncT1data><len><val>
                DFDF38 <EncT2status><len><val>
                DFDF39 <EncT2data><len><val>
                DFDF3A <EncT3status><len><val>
                DFDF3B <EncT3data><len><val>
                DFDF3C <Encrypted Magneprint Data><len><val>
                DFDF3D <MS2.0 Status><len><val> (MS2.0 Only)
                DFDF43 <Magneprint Status Data><len><val>
                DFDF50 (MSR KSN Data)<len><val> /*sent in the
clear*/
                    DFDF51 (MSR EncryptionType)<len><val>
                    F5<len> /* container tag for encrypted PIN data
(normally debit card) */
                        99 (Encrypted PIN DATA)<len><val>
                        DFDF41 (PIN KSN Data)<len><val>
                        DFDF42 (PIN EncryptionType)<len><val>

<Padding to force F9 plus padding to be a multiple of 8 bytes>

<Four byte CBC-MAC>

```

TLV data object FA contains the non-encrypted ARQC message.

The device calculates the CBC-MAC using the *Message Authentication, request or both ways* variant of the current MSR DUKPT working key used in the relevant transaction.

#### 3.6.3.2.2 SRED ARQC Request (SRED Only, MAC-MSR)

On SRED devices, the device sends ARQC messages using **Report 0x29 - Send Big Block Data to Host** with a message structured like the example below. The host may also customize the contents of the messages using tag DFDF02.

For details about using the general F9 MAC structure, see section **3.3 About Message Authentication Codes (“MAC-AMK” or “MAC-MSR”)**. Information specific to this message is provided after the example.

```
AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
  DFDF54(MAC KSN)<len><val>
  DFDF55(MAC Encryption Type)<len><val>
  DFDF25(IFD Serial Number)<len><val>
  FA<len> /* container for generic data */
    70<len> /*container for ARQC */
      DFDF53<len><value> /*fallback indicator */
      5F20<len><value> /*cardholder name */
      5F30<len><value> /*service code */
      DFDF4D<len><value> /* Mask T2 ICC Data */
      DFDF52<len><value> /* card type */
      F8<len> /*container tag for encryption */
        DFDF59<len><val> /* Encrypted Data
Primitive; decrypt data to read tags */
        DFDF56<len><val> /* Encrypted Transaction
Data KSN */>
        DFDF57<len><val> /* Encrypted Transaction
Data Encryption Type */>
        DFDF58<len><val> /* # of padding bytes
added to DFDF59 value to force length to a multiple of 8 bytes */
<Padding to force F9 plus padding to be a multiple of 8 bytes>
<Four byte CBC-MAC>
```

TLV data object F8 is an encrypted data container wrapping the encrypted ARQC message in nested data object DFDF59, plus supporting information as clear text in other tags.

The device encrypts the Value inside data container DFDF59 using the data variant of the current MSR DUKPT working key used in the relevant transaction. As a requirement for using the DUKPT TDES encryption algorithm, the device pads it so the length of its value is a multiple of 8 bytes. The device uses tag DFDF58 to report how many bytes of tag DFDF59 are padding. DFDF59 contains the following after the host decrypts it:

```
FC<len> /* container for encrypted generic data */
  F4<len> /* container tag for encrypted MSR
data */>
    DFDF36 <EncT1status><len><val>
    DFDF37 <EncT1data><len><val>
```

```
DFDF38 <EncT2status><len><val>
DFDF39 <EncT2data><len><val>
DFDF3A <EncT3status><len><val>
DFDF3B <EncT3data><len><val>
DFDF3C <Encrypted Magneprint
Data><len><val>
Only)
DFDF3D <MS2.0 Status><len><val> (MS2.0
DFDF43 <Magneprint Status
Data><len><val>
DFDF50 (MSR KSN Data)<len><val> /*sent
in the clear*/
DFDF51 (MSR EncryptionType)<len><val>
F5<len> /* container tag for encrypted PIN
data */
99 (Encrypted PIN DATA)<len><val>
DFDF41 (PIN KSN Data)<len><val>
DFDF42 (PIN EncryptionType)<len><val>
<Padding to force DFDF59 plus padding to be a
multiple of 8 bytes>
```

TLV data container F5 contains Encrypted PIN data using ISO Format 0 in nested data object 99, plus supporting information to decrypt it. The host should use the current PIN DUKPT working key specified in the supporting information.

The device calculates the CBC-MAC using the *Message Authentication, request or both ways* variant of the current MSR DUKPT working key used in the relevant transaction.

#### 3.6.3.3 Command 0xA2 Completion

When this command completes (card read OK, transaction finished, ICC problems, command cancelled, cardholder cancels, or timeout):

- 1) The device clears all sensitive data buffers and sends a **Report 0x22 - Card Status Report** to the host, and will continue to show **THANK YOU** on the display.
- 2) If the report indicates the cardholder has pressed the **Cancel** button, the host should abort the transaction.
- 3) If the Card Status and Operation Status are both OK, the host should send a request to get the EMV card data with **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**. It may call this multiple times if necessary.
- 4) The host should wait for as long as the solution design requires **THANK YOU** to show on the display.
- 5) The host should send the device **Command 0x02 - End Session** to clear **THANK YOU** from the display and make the device ready for the next transaction.

### 3.6.4 Command 0xA4 - Acquirer Response ARPC (MAC-MSR)

If a chip card and the device decide to handle a transaction online, the device will send an ARQC request to the host. The host should use this command to respond to the request, and to provide the response from the acquirer/issuer.

The ARPC response should be staged using big block data, and should include one or more of the following EMV data tags:

- ‘8A’ = response code
- ‘91’ = ARPC
- ‘71’ = Issuer Script Template 1
- ‘72’ = Issuer Script Template 2

The host should first send **Command 0x10 - Send Big Block Data to Device** to the device to send the Acquirer Response Data with the device serial number and signed with the current MSR MAC variant key. The data would be similar to the example below. Note the host must use the same KSN that was used in the ARQC request. Use container F9 for the MAC structure, use F8 within FA for passing the encrypted ARQC message, use MAC variant of MSR DUKPT key. This example assumes an acquirer host response of 70 04 8A 02 30 30:

```
AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
    DFDF54 (MAC KSN)<len><val>
    DFDF55 (Mac Encryption Type)<len><val>
    DFDF25 (IFD Serial Number)<len><val>
FA<len> /* Container for generic data */
    70 04 8A 02 30 30
    (ARQC padding, if any, to be a multiple of 8 bytes)
CBC-MAC (4 bytes, use MAC variant of MSR DUKPT key that was used in
ARQC request, from message length up to and including ARQC padding, if
any
```

The device can be factory configured to disable CBC-MAC verification (see **Command 0x09 - Set / Get Device Configuration**). In this mode, the MAC structure is still required, but the 4-byte CBC-MAC field can be filled with any values. Tag DFDF40 can be added to the ARQC request to indicate if CBC-MAC is checked by the device; a value of 0x80 means it is checked, 0x00 means not checked.

After sending the data, the host should issue the following command:

**Table 3-65 - Usage Table for Command 0xA4**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xA4							
Byte 1..12	Reserved							

### 3.6.5 Command 0xA8 - Get Kernel Info

This command directs the device to send the requested kernel information to the host.

**Table 3-66 - Usage Table for Command 0xA8 Set Mode**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xA8							
Byte 1	Kernel Info ID (see <b>Table 3-68</b> )							

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the host should call command 0xA8 in **Get** mode to retrieve a response in the following format:

**Table 3-67 - Usage Table for Command 0xA8 Get Mode**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xA8							
Byte 1	Kernel Info ID (see <b>Table 3-68</b> )							
Byte 2	Data length							
Byte 3..63	Block data							

**Table 3-68 - 0xA8 Kernel Info IDs**

Info ID	Description
0x00	Version - L1 Kernel
0x01	Version - L2 Kernel LIB
0x03	Version - L2 HAL
0x04	Version - S/W LIB
0x05	Reserved
0x06	Reserved
0x07	Reserved
0x08	Reserved
0x09	Reserved
0x0A	Reserved
0x10	Checksum/Signature - L1 Kernel
0x11	Checksum/Signature - L2 Kernel LIB
0x12	Checksum/Signature - L2 Kernel Configuration
0x13	Checksum/Signature - L2 HAL
0x14	Checksum/Signature - S/W LIB



Info ID	Description
0x1F	Checksum/Signature - L2 Kernel. This is the sum of all the checksums needed for the L2 Kernel and is the only value that should be monitored for L2 testing.
0x20	Reserved
0x21	Reserved
0x22	Reserved
0x23	Reserved
0x24	Reserved
0x25	Reserved
0x26	Reserved

### 3.6.6 Command 0xAB - Request EMV Transaction Data (MAC-MSR)

This command directs the device to send merchant data and pre-defined EMV batch data tags to the host, and for unsuccessful transactions can be used to send pre-defined reversal data. It is normally used by the host for data capture. The host should first successfully complete **Command 0xA2 - Start EMV Transaction**. The set of tags used during a given EMV transaction is a combination of the tags defined in the EMV specification and the tags that are specific to the kernel being used for the transaction. See the specifications for the payment brands for details. For EMV, see *EMV Integrated Circuit Card Specifications for Payment Systems v4.3*.

If an error occurs, the device will terminate the command and report the error in ACKSTS of **Command 0x01 - Response ACK**. For a full list of error codes, see **Appendix C Status and Message Table, Table 3-76**.

If no error occurs, the device will send **Report 0x29 - Send Big Block Data to Host** to the host. All data sent will be encrypted (SRED) with the DATA variant MSR key and signed using the MAC variant of the MSR key. The device serial number will also be part of the message.

**Table 3-69 - Usage Table for Command 0xAB**

Bit	7	6	5	4	3	2	1	0
Byte 0	0xAB							
Byte 1..4	0x00 (Reserved)							

EMV transaction batch data coming from a non-SRED device will begin with a two-byte header in big-endian form (MSB first) that contains the expected total length of the response data (all blocks 1..n), excluding data padding and CBC-MAC. Use container F9 for the MAC structure, FA for passing non-encrypted batch data message, use MAC variant of MSR DUKPT key. The general structure is as follows:

```

AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
    DFDF54 (MAC KSN) <len><val>
    DFDF55 (MAC Encryption Type) <len><val>
    
```

```
DFDF25(IFD Serial Number)<len><val>
FA<len> /* container for generic data */
  F0<len> /* Transaction Results */
    F1<len> /* container for Status Data */
    ... /* Status Data tags */
    F2<len> /* container for Batch Data */
    ... /* Batch Data tags */
    F3<len> /* container for Reversal Data, if any */
    ... /* Reversal Data tags */
    F7<len> /* container for Merchant Data */
    ... /* < Merchant Data tags */
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, use MAC variant of MSR DUKPT key)
```

EMV transaction batch data coming from an SRED device will begin with a two-byte header in big-endian form (MSB first) that contains the expected total length of the data for the command (all blocks 1..n), excluding data padding and CBC-MAC. Use container F9 for the MAC structure, use F8 within FA for passing encrypted batch data message, use MAC variant of MSR DUKPT key. The structure is as follows:

```
AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */
F9<len> /* container for MAC structure and generic data */
  DFDF54(MAC KSN)<len><val>
  DFDF55(MAC Encryption Type)<len><val>
  DFDF25(IFD Serial Number)<len><val>
  FA<len> /* container for generic data */
    F0<len> /* Transaction Results */
      F1<len> /* container for Status Data */
      ... /* Status Data tags */
      F8<len> /* container tag for encryption */
      DFDF59(Encrypted Data Primitive)<len><Encrypted
Data val (Decrypt data to read tags)>
      DFDF56(Encrypted Transaction Data KSN)<len><val>
      DFDF57(Encrypted Transaction Data Encryption
Type)<val>
      DFDF58(# of bytes of padding in DFDF59)<len><val>
      F7<len> /* container for Merchant Data */
      ... /* < Merchant Data tags */
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, use MAC variant of MSR DUKPT key)
```

The value inside tag DFDF59 is encrypted and contains the following after decryption:

```
FC<len> /* container for encrypted generic data */
F2<len> /* container for Batch Data */
  ... /* Batch Data tags */
F3<len> /* container for Reversal Data, if any */
  ... /* Reversal Data tags */
```

The following tables provide details about the data format. For an explanation of the “Format” columns, see the definitions in *EMV 4.3 Book 3*.

For more information about big blocks and TLV format, see section **3.1 About Big Block Data and TLV Format**. For an example of how to interpret the device's response to this command, see section **A.2 How to Parse Encrypted Big Block EMV Data From An SRED Device**.

**Table 3-70 - Big Block Response to Command 0xAB - Status Data Container (F1)**

Tag	Description	Source	Format	Length (decimal)
DFDF1A	Transaction Status 0x00 = Accept 0x01 = Decline 0x02 = Error 0x10 = Cancelled by Host 0x11 = Confirm Amount No 0x12 = Confirm Amount Timeout 0x13 = Confirm Amount Cancel 0x14 = MSR Select Credit 0x15 = MSR Select Debit 0x16 = MSR Select Credit/Debit timeout 0x17 = MSR Select Credit/Debit cancel 0x1B = PIN entry Cancelled by Host 0x1C = PIN entry timeout 0x1D = PIN entry Cancelled by Cardholder 0x1E = Manual Selection Cancelled by Host 0x1F = Manual Selection timeout 0x20 = Manual Selection Cancelled by Cardholder 0x21 = Waiting For Card Cancelled by Host 0x22 = Waiting For Card timeout 0x23 = Waiting For Card Cancelled by Cardholder 0x24 = Waiting For Card ICC Seated 0x25 = Waiting For Card MSR Swiped 0xFF = Unknown	Device	b	1
DFDF1B	Additional Transaction Information 0x00 = No additional information 0x31 = EMV Application not selected 0x32 = Error transaction in progress 0x33 = Error invalid PSE format 0x34 = EMV Terminal application list is empty 0x35 = Candidate list is empty 0x36 = No transaction 0x37 = No common EMV applications 0x38 = Transaction canceled 0x39 = Aid parse error 0x3A = Code table index not found 0x3B = Error no more record 0x3C = EMV e overflow [sic.]	Device	b	4

**Table 3-71** shows the factory default list of tags the device will include in the Batch Data Container (tag F2). This can be re-configured by changing the list of tags stored in tag DFDF02 (see **Command 0xA1 - Access EMV Tags**).

**Table 3-71 - Big Block Response to Command 0xAB - Batch Data Container (F2 [Default Tags Shown])**

Tag	Description	Source	Format	Length (decimal)
82	EMV Application Interchange Profile	Card	b	2
8E	CVM list	Card	b	0-252
5F24	Date after which the EMV Application expires	Card	n	3
5F25	Date from which the EMV Application can be used	Card	n	3
9F06	Indicates the EMV Application as described in <i>ISO/IEC 7816-5</i>	Device	b	5-16
9F07	Indicates issuer's specified restrictions on the geographic usage and services allowed for the EMV Application	Card	b	2
9F0D	Specifies the issuer's conditions that cause a transaction to be rejected if it might have been approved online, but the terminal is unable to process the transaction online	Card	b	5
9F0E	Specifies the issuer's conditions that cause the denial of a transaction without attempt to go online	Card	B	5
9F0F	Specifies the issuer's conditions that cause a transaction to be transmitted online	Card	B	5
9F10	Contains proprietary EMV application data for transmission to the issuer in an online Transaction	Card	b	0-32
9F26	Cryptogram returned by the ICC in response to the GENERATE AC command	Card	b	8
9F27	Indicates the type of cryptogram and the actions to be performed by the terminal	Card	b	1
9F36	Counter maintained by the EMV application in the ICC (incrementing the ATC is managed by the ICC)	Card	b	2
95	TVR	Device	b	5
9B	TSI	Device	b	2
9C	Transaction Type	Device	b	2

Tag	Description	Source	Format	Length (decimal)
9F33	Terminal Capabilities	Device	b	3
9F34	Indicates the results of the last CVM performed	Device	b	3
9F37	Value to provide variability and uniqueness to the generation of a cryptogram	Device	b	4
9F40	Additional Terminal Capabilities	Device	b	5
DFDF70	TAC-default (Terminal Action Codes)	Device	n	5
DFDF71	TAC-Offline (Terminal Action Codes)	Device	n	5
DFDF72	TAC-Online (Terminal Action Codes)	Device	n	5
9F5B	Issuer Script Results	Device	b	0-128

The Merchant Data (F7) Container is included in the response to **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**, which is normally used by the host for receipt printing. The tags in the F7 container are not programmable.

**Table 3-72 - Big Block Response to Command 0xAB - Merchant Data Container (F7)**

Tag	Description	Source	Format	Length (decimal)
DFDF30	Masked T1 Status	Device	b	1
DFDF31	Masked T1	Device	b	Var
DFDF32	Masked T2 Status	Device	b	1
DFDF33	Masked T2	Device	b	Var
DFDF34	Masked T3 Status	Device	b	1
DFDF35	Masked T3	Device	b	Var
DFDF40	Signature Required	Device	b	1
5F25	EMV Application Effective Date	Card	N6	3
5F24	EMV Application Expiration Date	Card	N6	3
89	Authorization Code	Device	b	6
5F2A	Transaction Currency Code	Card	N3	2
9F02	Amount, authorized	Device	N12	6
9F03	Amount, other	Device	N12	6
9F06	AID - terminal	Device	b	Var 5-16
9F12	EMV Application Preferred Name	Card	ans	Var 1-16
9F1C	Terminal ID	Device	An 8	8
9F39	POS Entry Mode	Device	N2	1

Tag	Description	Source	Format	Length (decimal)
9C	Transaction Type	Device	N2	1
9F34	Indicates the results of the last CVM performed	Device	b	3
5F57	Account Type	Device	N2	1
5F34	PAN Sequence Number	Card	N2	1
5F20	Cardholder Name	Card	ans	Var 2-26
DFDF4D	Masked ICC Track 2 Data	Card	ans	Var 30-38

The Reversal Data (F3) Container may be included in the response to **Command 0xAB - Request EMV Transaction Data (MAC-MSR)** normally used by the host for reversal processing. This data is sent by the device for an online transaction where the acquirer response was to approve the transaction, but the final card decision was to decline. **Table 3-73** shows the factory default list of tags the device will include in the Reversal Data Container (tag F3).

**Table 3-73 - Big Block Response to Command 0xAB - Reversal Data Container (F3)**

Tag	Description	Source	Format	Length
82	EMV Application Interchange Profile	Card	b	2
9F36	Counter maintained by the EMV application in the ICC (incrementing the ATC is managed by the ICC)	Card	b	2
DFDF25	Terminal Serial Number	Device	an	8
9F10	Contains proprietary EMV application data for transmission to the issuer in an online Transaction	Card	b	0-32
9F5B	Issuer Script Results	Device	b	0-128
9F33	Terminal Capabilities	Device	b	3
9F35	Terminal Type	Device	n	1
95	TVR	Device	b	5
9F01	Uniquely identifies the acquirer within each payment system	Device	n	6
5F24	Date after which the EMV Application expires	Card	n	3
5A	Primary Account Number	Card	c	0-10
5F34	PAN Sequence Number	Card	n	3
8A	Authorization Code	Device	an	2
9F15	Merchant Category Code	Device	n	2
9F16	Merchant Identifier	Device	s	15
9F39	POS Entry Mode	Device	n	1

Tag	Description	Source	Format	Length
9F1A	Terminal Country Code	Device	n	2
9F1C	Terminal ID	Device	an	8
57	Track2 Equivalent Data	Card	b	0-19
9F02	Amount Authorized	Device	n	6
5F2A	Transaction Currency Code	Device	n	2
9A	Transaction Date	Device	n	3
9F21	Transaction Time	Device	n	3
9C	Transaction Type	Device	n	1

### 3.6.7 Command 0xAC - Merchant Bypass PIN Command

This command allows the host to bypass the PIN entry requirement during an EMV transaction (**Command 0xA2 - Start EMV Transaction**).

Table 3-74 - Usage Table for Command 0xAC

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

## Appendix A Examples

### A.1 How to Get MSR/PIN Data from the Device for a Bank Simulation

This section provides a byte-by-byte example of transmitting commands on various devices using a USB connection, an 802.11 wireless connection, an Ethernet connection, or an Apple 30-pin connection. All data shown in this section is in hexadecimal format. 802.11 wireless, Ethernet, and USB connections require bytes to be transmitted in least significant byte (little endian) order; iOS requires bytes to be transmitted in most significant byte (big endian) order, so iOS command strings will appear as the byte-by-byte reverse of the others.

- 1) Host sends out **Command 0x03 - Request Swipe Card** to the device, which expands to the following bytes:
  - a) C0: Ethernet packet header (802.11 wireless and Ethernet only)
  - b) 01: Execute command in Get mode (802.11 wireless, Ethernet, & iOS only)
  - c) 03: Command ID (03=**Command 0x03 - Request Swipe Card**)
  - d) 20: Wait time (20=32 seconds)
  - e) 00: Display message ID (00=swipe card/idle)
  - f) 01: Beep prompt tone for card swipe (01=one beep)
  - g) C0: Ethernet packet terminator (802.11 wireless and Ethernet only)

#### Sample command data of Command 0x03 - Request Swipe Card

03 20 00 01	USB format of command
C0 01 03 20 00 01 C0	802.11 wireless and Ethernet format of command
01 03 20 00 01	iOS format of command (Note MSB order)

- 2) Device sends back **Command 0x01 - Response ACK** to the host, which expands to the following bytes. If **Command 0x03 - Request Swipe Card** had failed (i.e. ACK status not = 00), the device would not have returned a device state input report to the host:
  - a) C0: Ethernet packet header (802.11 wireless and Ethernet only)
  - b) 02: Response to command (802.11 wireless and Ethernet only)
  - c) 01: Report ID (01=**Command 0x01 - Response ACK**)
  - d) 00: ACK status of **Command 0x03 - Request Swipe Card** (00=Command is good)
  - e) 03: Command ID of the command being ACKed (03=**Command 0x03 - Request Swipe Card**)
  - f) C0: Ethernet packet terminator (802.11 wireless and Ethernet only)

#### Sample response for Command 0x01 - Response ACK

01 00 03	USB format of command
C0 02 01 00 03 C0	802.11 wireless and Ethernet format of command
01 00 03	iOS format of command (Note MSB order)

- 3) The device prompts the cardholder to swipe his or her card, and sends **Report 0x20 - Device State Report** to the host, which expands to the following bytes:



- a) C0: Ethernet packet header (802.11 wireless and Ethernet only)
- b) 03: Unsolicited response (802.11 wireless and Ethernet only)
- c) 20: Report ID (20=**Report 0x20 - Device State Report**)
- d) 02: Device state (02=Wait for card)
- e) 08: Session state (08=Card data available)
- f) 40: Device status (40=Not authenticated)
- g) 47: Device cert status (PIN CRL, PIN CA cert, Device CA cert & Device cert exist)
- h) 07: Hardware status (Keypad calibrated, Mag Head programmed, Tamper sensors active)
- i) C0: Ethernet packet terminator (802.11 wireless and Ethernet only)

**Sample Report 0x20 - Device State Report**

20 02 08 40 47 07	USB format of command
C0 03 20 02 08 40 47 07 C0	802.11 and Ethernet format of command
20 02 08 40 47 07	iOS format of command (Note MSB order)

- 4) After the cardholder swipes the card, the device sends back **Report 0x22 - Card Status Report** to the host, which expands to the following bytes:
- a) C0: Ethernet packet header (802.11 wireless and Ethernet only)
  - b) 03: Unsolicited response (802.11 wireless and Ethernet only)
  - c) 22: Report ID (22=**Report 0x22 - Card Status Report**)
  - d) 00: Operation status (00=OK)
  - e) 00: Card status (00=OK)
  - f) 01: Card type (01=Financial card)
  - g) C0: Ethernet packet terminator (802.11 wireless and Ethernet only)

**Sample Report 0x22 - Card Status Report**

22 00 00 01	USB format of command
C0 03 22 00 00 01 C0	802.11 wireless and Ethernet format of command
22 00 00 01	iOS format of command (Note MSB order)

- 5) If the operation and the card status are OK, the host retrieves the card data from the device by issuing **Command 0x0A - Request MSR Data**, as shown:

**Sample Command 0x0A - Request MSR Data:**

0A 00	USB format of command
C0 01 0A 00 C0	802.11 wireless and Ethernet format of command
0A 00	iOS format of command (Note MSB order)

- 6) The device sends back **Command 0x01 - Response ACK** to the host.
- 7) The device sends back eight instances of **Report 0x23 - Card Data Report** to the host, which the host interprets as meaning the following (Note that additional headers for Ethernet are not shown):

- a) Track 1: 23 01 00 2F 0-0x2E bytes of data
  - b) Track 2: 23 02 00 1E 0-0x1D bytes of data
  - c) Track 3: 23 03 00 47 0-0x46 bytes of data
  - d) Encrypted Track1: 23 04 00 30 0-0x2F bytes of data
  - e) Encrypted Track2: 23 05 00 20 0-0x1F bytes of data
  - f) Encrypted Track3: 23 06 00 48 0-0x47 bytes of data
  - g) Encrypted MagnePrint: 23 07 00 38 0-0x37 bytes of data
  - h) KSN and MagnePrint Status: 23 63 00 0E 0-0x0D bytes of data
- 8) The device sends back another **Report 0x20 - Device State Report** to the host.
- 9) If the operation status and card status from **Report 0x22 - Card Status Report** sent previously are both OK, the host issues **Command 0x04 - Request PIN Entry**, which expands to the following bytes:
- a) C0: Ethernet packet header (802.11 wireless and Ethernet only)
  - b) 01: Execute command in Get mode (802.11 wireless, Ethernet, & iOS only)
  - c) 04: Command ID (04=**Command 0x04 - Request PIN Entry**)
  - d) 1E: Wait time for PIN entry (1E=30 seconds)
  - e) 00: PIN mode (00=Enter PIN)
  - f) 44: Max and min length of PIN (in this example, PIN must be exactly four characters)
  - g) 01: Prompt tone (01=One beep)
  - h) 01: PIN option (01=ISO3)
  - i) C0: Ethernet packet terminator (802.11 wireless and Ethernet only)

**Sample Command 0x04**

04 1E 00 44 01 01	USB format of command
C0 01 04 1E 00 44 01 01 C0	802.11 and Ethernet format of command
01 04 1E 00 44 01 01	iOS format of command (Note MSB order)

- 10) The device sends the host **Command 0x01 - Response ACK** if the command is successful.
- 11) The device sends the host **Report 0x24 - PIN Response Report** if PIN entry is successful.
- 12) The device sends the host another **Report 0x20 - Device State Report**.

## A.2 How to Parse Encrypted Big Block EMV Data From An SRED Device

This section provides an example of parsing encrypted EMV data sent from an SRED device in TLV format, in response to **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**. For more information about big blocks and TLV format, see section **3.1 About Big Block Data and TLV Format**.

In this case, the device sends big block data to the host in the big block data buffer via **Report 0x29 - Send Big Block Data to Host**. Upon assembling the incoming packets and stripping off the packet containers, the block of data looks like this:

```
01C7F98201C3DFDF540A9500030000000120039DDDFDF550182DFDF250898CE04611018
060DFA8201A0F082019CF105DFDF1A0100F882010EDDFDF598200F08569A27E2A2A9D7E
67A96624D10DBE3F366EC3F31C4072676FEF43213AF3C76ABE06A6E90F10E1650BE4EC
E9CF64E9143129F66B44E8C4A697CA5A0E319D933BF9BBC52B2DAF8FCC663354E2B0E5
45A5002F4A0C976E65DD23705AB36ECA78D6A6B99243F2C2B907A8F1F2A66D5558096D
7B1F91F1B6C06BF68841098EEABA502A57A3AA2F1344C4E405B86C3D64FB93E638D821
409493659966A247238109C0E117B669B74A5508261B8E8AFF3FFE68058C334B383D99
1EAE3C8F5594FBFB9118860FF67344F37DE54EA5F28BFECF8378072A9FAE3A61FEF132
54B6C7B2C1D0AF626E5A14F19C025B7CD1EF1456A31DDDFDF560A950003000000012003
9DDDFDF570180DFDF580106F782007FDFDF4001015F25031201015F24031401315F2A02
08269F02060000000002009F0607A00000000410109F1C0831313232333334349F3901
059C01009F34035E03005F5701305F3401005F2009544553542F43415244DFDF4D263B
353431333030303034303030313531333D30313134303030303030303030303030303030
303F0000000000000000C568ACEB
```

Because the calling command is tagged with “MAC,” its response should be interpreted using the information in section **3.3 About Message Authentication Codes (“MAC-AMK”)**. Specifically because this is batch data coming from an SRED device, the data is interpreted using section **3.6.6 Command 0xAB - Request EMV Transaction Data (MAC-MSR)**, so the data would be broken down as follows:

01C7 is the expected data length not including padding and CBC-MAC.

Per **Appendix D MagTek Custom EMV Tags** and section **3.6.6**, the next byte, F9, is a tag that indicates the beginning of a data object containing Message Authentication (MAC) structure and generic data.

According to *EMV Integrated Circuit Card Specifications for Payment Systems 4.3, Part IV, Annex B Rules for BER-TLV Data Objects*, the next three bytes, 8201C3, indicate the length of the value in data block F9; lengths can be either one byte long from 0x00 to 0x7F, or multiple bytes starting with 0x80 or higher, in which case the lower 7 bits of the first byte specify how many subsequent bytes will indicate the length of the value in the data object. In this case, 0x82 is greater than 0x7F, and the lower 7 bits equal 0x02, so the next 2 bytes 0x01C3 give the total length of the data block for tag F9. The value in data object F9 therefore consists of these 451 bytes:

```
DFDF540A9500030000000120039DDDFDF550182DFDF250898CE04611018060DFA8201A0
F082019CF105DFDF1A0100F882010EDDFDF598200F08569A27E2A2A9D7E67A96624D10D
BE3F366EC3F31C4072676FEF43213AF3C76ABE06A6E90F10E1650BE4ECE9CF64E91431
29F66B44E8C4A697CA5A0E319D933BF9BBC52B2DAF8FCC663354E2B0E545A5002F4A0C
976E65DD23705AB36ECA78D6A6B99243F2C2B907A8F1F2A66D5558096D7B1F91F1B6C0
6BF68841098EEABA502A57A3AA2F1344C4E405B86C3D64FB93E638D821409493659966
A247238109C0E117B669B74A5508261B8E8AFF3FFE68058C334B383D991EAE3C8F5594
FBFB9118860FF67344F37DE54EA5F28BFECF8378072A9FAE3A61FEF13254B6C7B2C1D0
```



The value of data object F0 (transaction results) begins with tag F1, which **Appendix D MagTek Custom EMV Tags** and section **3.6.6** indicate is status data. Its length is a single byte 0x05, giving the value DFDF1A0100.

The value of data object F0 (transaction results) continues with tag F8 (container tag for encryption), length 0x82010E, giving a 270 byte value:

```
DFDF598200F08569A27E2A2A9D7E67A96624D10DBE3F366EC3F31C4072676FEF43213A
F3C76ABE06A6E90F10E1650BE4ECE9CF64E9143129F66B44E8C4A697CA5A0E319D933B
F9BBC52B2DAF8FCC663354E2B0E545A5002F4A0C976E65DD23705AB36ECA78D6A6B992
43F2C2B907A8F1F2A66D5558096D7B1F91F1B6C06BF68841098EEABA502A57A3AA2F13
44C4E405B86C3D64FB93E638D821409493659966A247238109C0E117B669B74A550826
1B8E8AFF3FFE68058C334B383D991EAE3C8F5594FBFB9118860FF67344F37DE54EA5F2
8BFECF8378072A9FAE3A61FEF13254B6C7B2C1D0AF626E5A14F19C025B7CD1EF1456A3
1DDDFDF560A9500030000000120039DDDFDF570180DFDF580106
```

The value of data object F8 begins with tag DFDF59 (encrypted data primitive), length 0x8200F0, giving a 240 byte value containing encrypted data:

```
8569A27E2A2A9D7E67A96624D10DBE3F366EC3F31C4072676FEF43213AF3C76ABE06A6
E90F10E1650BE4ECE9CF64E9143129F66B44E8C4A697CA5A0E319D933BF9BBC52B2DAF
8FCC663354E2B0E545A5002F4A0C976E65DD23705AB36ECA78D6A6B99243F2C2B907A8
F1F2A66D5558096D7B1F91F1B6C06BF68841098EEABA502A57A3AA2F1344C4E405B86C
3D64FB93E638D821409493659966A247238109C0E117B669B74A5508261B8E8AFF3FFE
68058C334B383D991EAE3C8F5594FBFB9118860FF67344F37DE54EA5F28BFECF837807
2A9FAE3A61FEF13254B6C7B2C1D0AF626E5A14F19C025B7CD1EF1456A31D
```

This data must be decrypted (described later) to read the remaining TLV data inside it.

After the encrypted data primitive, the value of data object F8 continues with:

```
DFDF560A9500030000000120039DDDFDF570180DFDF580106
```

This breaks down into tag DFDF56 (encrypted transaction data KSN), length 0x0A giving 10 byte value 9500030000000120039D, tag DFDF57 (encrypted transaction data encryption type), length 0x01, value 0x80 (DUKPT key), and tag DFDF58 (number of bytes padding in DFDF59), length 0x01, value 0x06.

To decrypt the value from data object DFDF59 above, use the data key variant and the KSN from DFDF56 above (9500030000000120039D) and the ANSI test key to perform 3DES-DUKPT decryption on the value. For details about the decryption algorithm, see **ANSI X9.24**. In this case, the value decrypts and parses as:

Tag	Length	Value
82 (EMV Application Interchange Profile)	0002	5800
8E (CVM)	0010	000000000000000042015E0342031F03



The value consists of tag F7 (merchant data tags), length 0x82007F, which is always unencrypted, and parses to the following:

Tag	Length	Value
DFDF40 (Signature or MAC Required)	0001	01
5F25 (EMV Application Effective Date)	0003	120101
5F24 (EMV Application Expiration Date)	0003	140131
5F2A (Transaction Currency Code)	0002	0826
9F02 (Amount Authorized)	0006	000000000200
9F06 (EMV Application AID)	0007	A0000000041010
9F1C (Terminal ID)	0008	3131323233333434
9F39 (POS Entry Mode)	0001	05
9C (Transaction Type)	0001	00
9F34 (CVM Results - Terminal)	0003	5E0300
5F57 (Account Type)	0001	30
5F34 (PAN Sequence Number)	0001	00
5F20 (Cardholder Name)	0009	544553542F43415244
DFDF4D (Mask T2 ICC Data)	0026	3B353431333030303034303030313531333D3 03131343030303030303030303030303030303030 3F ("TEST/CARD")

And the remaining data in the whole big block is:

```
00000000000000000000C568ACEB
```

...which consists of padding zeroes and the 4-byte CBC-MAC C568ACEB.

## Appendix B Terminology

This appendix provides definitions of common terms used in this document.

**Table 3-75 - Common Terms**

Term	Definition
AAC	Application Authentication Cryptogram
AAMVA	American Association of Motor Vehicle Administrators
ACK	Acknowledge
AES	Advanced Encryption Standard
AMK	Acquirer Master Key
AP	[802.11 Wireless] Access Point
API	Application Programming Interface
ARC	Authorization Response Code
ARQC	Authorization Request Cryptogram
ARPC	Authorization Response Cryptogram
APDU	Application Protocol Data Unit
ATR	Answer To Reset
BIN	Bank/Issuer Identification Number
Bluetooth LE	Bluetooth Low Energy
BPK	Battery Protected Keys
CA	Certificate Authority
CAPK	Certificate Authority Public Key
CBC	Cipher Block Chaining
CDA	Combined DDA/Application Cryptogram Generation
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CVC	Card Verification Code
CVM	Cardholder Verification Method
DDA	Dynamic Data Authentication
DER	Distinguished Encoding Rules
DES	Data Encryption Standard. An algorithm developed in the 1970s by IBM Corporation, since adopted by the US government and ANSI (the American National Standards Institute) as the encryption standard for financial institutions.



Term	Definition
DLL	Dynamically Linked Library
DOL	Data Object List
DUKPT	Derived Unique Key Per Transaction. A key management scheme in which a unique key is used for every transaction
CBC	Cipher Block Chaining
ECDB	Encrypted Cardholder Data Block
EMV[co]	Europay MasterCard Visa [company]
EPB	Encrypted PIN Block
GATT	Generic ATtribute Profile, a general specification for sending and receiving short pieces of data known as "attributes" over a Bluetooth LE link.
HAL	Hardware Abstraction Layer
HID	Human Interface Device
I <sup>2</sup> C	Inter-Integrated Circuit
iAP	iPod Accessory Protocol
iOS	Apple device operating system
ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
ISO	International Standards Organization
Key Injection	A secure operation whereby an encryption key is injected into a device
KCV	Key Check Value
KSN	Key Serial Number
LCD	Liquid Crystal Display
LSB	Least Significant Byte, Least Significant Bit
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.

Term	Definition
MS2.0	MagneSafe 2.0, MagTek's proprietary method of encrypting data while preserving the essential format of the unencrypted data, such as length and character set.
MSB	Most Significant Byte, Most Significant Bit
MSR	Magnetic Stripe Reader
PAN	Personal Account Number, which is most commonly recognized as the 16-digit account number associated with a card.
PCI DSS	Payment Card Industry Data Security Standards
PCI PED	Payment Card Industry PIN Entry Device
PED	Pin Encryption Device, the generic term for the class of devices that includes IPAD, DynaPro, and DynaPro Mini.
PIN	Personal Identification Number
PKI	Public Key Infrastructure. An arrangement that binds public keys with respective user identities by means of a certificate authority.
RFU	Reserved for Future Use
RID	Registered Application Provider Identifier
RSA	Rivest-Shamir-Adleman. A highly secure cryptography method by RSA Data Security, Inc., Redwood City, CA ( <a href="http://www.rsa.com">www.rsa.com</a> ), which uses a two-part key: The private key is kept by the owner, and the public key is published.
RNG	Random Number Generator
RTC	Real Time Clock
SPI	Serial Peripheral Interconnect
SDA	Static Data Authentication
SDRAM	Synchronous Dynamic Random Access Memory
SRAM	Static Random Access Memory
SRED	Secure Reading and Exchange of Data
TAC	Terminal Action Codes
TC	Transaction Certificate
TDES	Triple Data Encryption Standard
TRSM	Tamper-Resistant Security Module
TSI	Transaction Status Information

Term	Definition
TVR	Transaction Verification Results
UART	Universal Asynchronous Receive Transmit
USB	Universal Serial Bus
USIP	Universal Secure Integrated Platform

## Appendix C Status and Message Table

Table 3-76 - Status and Message Codes

Status/Msg	Value
Operation Status	0x00 = OK / Done 0x01 = Cardholder Cancel 0x02 = Timeout 0x03 = Host Cancel 0x04 = Verify fail 0x05 = Keypad Security 0x06 = Calibration Done 0x07 = Write with duplicate RID and index 0x08 = Write with corrupted Key 0x09 = CA Public Key reached maximum capacity 0x0A = CA Public Key read with invalid RID or Index

Status/Msg	Value
ACK Status ("ACKSTS")	0x00 = OK / Done 0x15 - RID error / Index not found 0x80 = Device Error - A device error or tamper has been detected, the device certificate is missing or has been changed, or signature is not correct. 0x81 = Device not Idle 0x82 = Data Error or Bad Parameter(s) - The command contains bad parameters. For example, in a big block data transfer, the parameters in any block 1 through n data packet don't match (or don't follow) the previous data packet's parameters; it may also indicate a bad CBC-MAC ACKSTS, wrong serial number, or a bad key. 0x83 = Length Error - The data size is 0 or is larger than the available buffer size, or a data packet is incomplete, or MagTek device OID of the certificate doesn't match the predefined OID 0x84 = PAN Exists 0x85 = No Key or Key is incorrect - No authentication key, no Acquirer Master Key, or an invalid key is found in the device 0x86 = Device busy 0x87 = Device Locked - More than 120 PINs were entered within one hour, or there have been three authentication failures, or a previous call to <b>Command 0x09 - Set / Get</b> Device Configuration locked the device's configuration. 0x88 = Auth required 0x89 = Bad Auth - Host has sent an incorrect authentication token (e.g., the decrypted random token or device serial number doesn't match the device's current values) 0x8A = Device not Available, Device Status is not OK, Touchscreen is not connected or doesn't exist, or Authentication challenge token has timed out (i.e. is not used within 5 minutes) 0x8B = Amount Needed - If PIN amount is required, no amount has been sent 0x8C = Battery is critically low (refuse new transactions and host commands) 0x8D = Device is resetting (refuse new transactions and host commands) 0x90 = Cert non-exist - For unbind/rebind/key injection, the associated certificate doesn't exist 0x91 = Expired (Cert/CRL) 0x92 = Invalid (Cert/CRL/Message) 0x93 = Revoked (Cert/CRL) 0x94 = CRL doesn't exist 0x95 = Cert exists 0x96 = Duplicate KSN/Key
Display Message	0x00 = Hands Off 0x01 = Approved 0x02 = Declined 0x03 = Cancelled 0x04 = Thank You 0x05 = PIN Invalid 0x06 = Processing 0x07 = Please Wait

Status/Msg	Value
EMV Message	<b>EMV 4.3 BOOK 4 Section 11.2 “Standard Messages.”</b> 0x01 = (AMOUNT) 0x02 = (AMOUNT) OK? 0x03 = APPROVED 0x04 = CALL YOUR BANK 0x05 = CANCEL OR ENTER 0x06 = CARD ERROR 0x07 = DECLINED 0x08 = ENTER AMOUNT 0x09 = ENTER PIN 0x0A = INCORRECT PIN 0x0B = INSERT CARD 0x0C = NOT ACCEPTED 0x0D = PIN OK 0x0E = PLEASE WAIT 0x0F = PROCESSING ERROR 0x10 = REMOVE CARD 0x11 = USE CHIP READER 0x12 = USE MAGSTRIPE 0x13 = TRY AGAIN
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 = Transaction Type 0x01 = Amount OK
Card Message	0x00 = Swipe Card / Idle alternating 0x01 = Swipe Card 0x02 = Please Swipe Card 0x03 = Please Swipe Again
Buzzer	0x00 = None 0x01 = Single Beep 0x02 = Double Beep
Amount Type	0x00 = Credit 0x01 = Debit

Status/Msg	Value							
Device State	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 0x09 = Wait Cardholder Entry 0x0A = Chip Card 0x0B = ICC Kernel Test 0x0C = EMV Transaction 0x0D = Show PAN							
Card Type	0x00 = Other 0x01 = Financial 0x02 = AAMVA 0x03 = Manual 0x04 = Unknown 0x05 = ICC 0x06 = Contactless ICC - EMV 0x07 = Financial MSR + ICC (Byte 2 bit 2 of device configuration must be set) 0x08 = Contactless ICC - MSD							
Card Status	0x00 = OK Otherwise, for each track, the possible values are listed below:							
	Value 0 = no error Value 1 = error detected							
	Bit 7	6	5	4	3	2	1	0
0	0	0	ICC	Track 3	Track 2	Track 1	0	
Key Mask	For each key, the possible values are listed below: Value 0 = the indicated key was not pressed Value 1 = the indicated key was pressed							
	Bit 7	6	5	4	3	2	1	0
	0	0	0	0	Enter	Right	Middle	Left

Status/Msg	Value																							
Device Status	<p>0x00 = OK. Otherwise, the possible values are listed below:</p> <p>Bits 0..1 = PIN Key Status:                      00 = PIN Key OK                      01 = PIN Key Exhausted                      10 = No PIN Key                      11 = PIN Key Not Bound</p> <p>Bits 2..3 = MSR Key Status:                      00 = MSR Key OK                      01 = MSR Key Exhausted                      10 = No MSR Key                      11 = MSR Key Not Bound</p> <p>Bit 4 = Tamper:                      0 = Normal                      1 = Tamper Detected</p> <p>Bit 5 = 0</p> <p>Bit 6 = Authentication Status:                      0 = Not Authenticated                      1 = Authenticated</p> <p>Bit 7 = Device Error Status:                      1 = Device Error (Can be cleared by calling <b>Command 0x02 - End Session</b>)</p>																							
Session State	<p>The possible values are listed below:</p> <p>Pwr Chg:                      1 = Power Change Occurred (occurs on Power up or after a USB resume)</p> <p>Card Data:                      1 = Card Data Available</p> <p>MSR PAN:                      1 = PAN Parsed from Card</p> <p>EXPAN:                      1 = External PAN Sent</p> <p>Amt:                      1 = Amount sent</p> <table border="1" data-bbox="393 1654 1461 1780"> <thead> <tr> <th data-bbox="393 1654 490 1696">Bit 7</th> <th data-bbox="496 1654 587 1696">6</th> <th data-bbox="594 1654 685 1696">5</th> <th data-bbox="691 1654 782 1696">4</th> <th data-bbox="789 1654 880 1696">3</th> <th data-bbox="886 1654 977 1696">2</th> <th data-bbox="984 1654 1075 1696">1</th> <th data-bbox="1081 1654 1172 1696">0</th> </tr> </thead> <tbody> <tr> <td data-bbox="393 1705 490 1780">Pwr Chg</td> <td data-bbox="496 1705 587 1780">0</td> <td data-bbox="594 1705 685 1780">0</td> <td data-bbox="691 1705 782 1780">0</td> <td data-bbox="789 1705 880 1780">Card Data</td> <td data-bbox="886 1705 977 1780">MSRPAN</td> <td data-bbox="984 1705 1075 1780">EXPAN</td> <td data-bbox="1081 1705 1172 1780">Amt</td> </tr> </tbody> </table>								Bit 7	6	5	4	3	2	1	0	Pwr Chg	0	0	0	Card Data	MSRPAN	EXPAN	Amt
Bit 7	6	5	4	3	2	1	0																	
Pwr Chg	0	0	0	Card Data	MSRPAN	EXPAN	Amt																	
	<p>0 = Certificate does not exist in the device                      1 = Certificate exists in the device</p>																							



**Appendix C - Status and Message Table**

---

Status/Msg	Value							
	Bit 7	6	5	4	3	2	1	0
Device Certificate Status	MSR CRL	PIN CRL	0	Mfg Unbind	MSR CA	PIN CA	Device CA	Device Cert
	0 = False 1 = True							
Hardware Status	Bit 7	6	5	4	3	2	1	0
	0	.IE3 only	SRED	Reserved	Reserved	Reserved	MagHead Programmed (IntelliHead Only)	Tamper Sensors Active

## Appendix D MagTek Custom EMV Tags (EMV Only)

In addition to the standard EMV tags documented in *EMV 4.3, Book 3, Annex A*, MagTek provides additional custom tags with the device. These are used with **Command 0xA1 - Access EMV Tags**, **Command 0xA2 - Start EMV Transaction**, and **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**. The custom tags are listed in **Table 3-77**. The characters used in the “Format” column are described in *EMV 4.3, Book 4, Section 4.3*.

**Table 3-77 - MagTek Custom EMV Tags**

Tag	Description	Default (HEX)	Format	Length
F0	Container for Status, Batch, Reversal, Merchant	*	b	var
F1	Status Data (Constructed Data Object)	*	b	var
F2	Batch Data (Constructed Data Object)	*	b	var
F3	Reversal Data (Constructed Data Object)	*	b	var
F4	Encrypted MSR Data (Constructed Data Object)	*	b	var
F5	Encrypted PIN Data (Constructed Data Object)	*	b	var
F7	Container for Merchant Data	*	b	var
F8	Container for Encrypted Data	*	b	var
F9	Container For Message Authentication (MAC)	*	b	var
FA	Container for Generic Data	*	b	var
FB	Container for BIN table	*	b	var
FC	Container for Encrypted Generic Data	*	b	var
DFDF00	Random number for random online transaction	8D	b	1
DFDF01	Revoked Certificate Lists (Not supported)	A0 00 00 00 03 96 FF FF FF A0 00 00 00 04 96 FF FF FF A0 00 00 00 05 96 FF FF FF	b	var up to 72
DFDF02	Authorization Request Tags (ARQC)	9F 03 9F 26 82 5A 5F 34 9F 36 9F 1A 95 9F 02 5F 2A 9A 9C 9F 37 9F 10	b	var up to 158
DFDF03	Advice Tags (Not supported)	5A	b	var up to 161
DFDF04	Financial Request Tags (ARPC) (Not supported)	91 71 72 9F 01 89 8A	b	var up to 433

Tag	Description	Default (HEX)	Format	Length
DFDF05	Reversal Tags	82 9F 36 9F 1E 9F 10 9F 5B 9F 33 9F 35 95 9F 01 5F 24 5A 5F 34 8A 9F 15 9F 16 9F 39 9F 1A 9F 1C 57 9F 02 5F 2A 9A 9F 21 9C	b	var up to 123
DFDF06	Authorization Response Tags	8A 91	b	var up to 123
DFDF07	Certification Validation Table (Not supported)	00	b	1
DFDF10	Threshold Value for Biased Random Selection	00 00 00 00 40 00	N	6
DFDF11	Target Percentage to be used for Random Selection (0-99 decimal, or 0-63 hex)	32	b	1
DFDF12	Maximum Target Percentage to be used for Biased Random Selection (0-99 decimal, or 0-63 hex)	46	b	1
DFDF13	Default CVM for the EMV application	01	N	1
DFDF14	Socket Timeout	00 00 0B B8	b	4
DFDF15	Socket Retries	00 00 00 01	b	4
DFDF16	Issuer Script max size	00 00 00 80	N	4
DFDF17	Batch Data Tags	82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 10 9F 26 9F 27 9F 36 95 9B 9C 9F 33 9F 34 9F 37 9F 40 FF 0D FF 0E FF 0F 9F 5B	b	var up to 438
DFDF19	Default Terminal Language	65 6E	b	2
DFDF1A	Transaction Status - Part of F1 container for status. Always present.	*	b	1
DFDF1B	Additional Transaction Information - Part of F1 container for status. May not be present.	*	b	4
DFDF20	Terminal Features	57	b	1
DFDF21	Number of EMV Applications	0A	b	1
DFDF22	PSE Name	31 50 41 59 2E 53 59 53 2E 44 44 46 30 31	b	14
DFDF23	ASI (Application Select Indicator)	01	b	1

Tag	Description	Default (HEX)	Format	Length
DFDF24	Requested Transaction Type	*	b	1
DFDF25	Unique and permanent serial number assigned to the IFD by the manufacturer	USIP SN	b	8
DFDF26	Device & EMV Application Database Label		b	16
DFDF27	Device & EMV Application Database Checksum	Read-only, calculated by device	b	20
DFDF28	CAPK Database Label		b	16
DFDF29	CAPK Database Checksum	Read-only, calculated by device	b	20
DFDF2D	Supported Terminal Languages	65 6e 66 72 69 74 64 65 65 73	b	10
DFDF30	Masked T1 Status	*	b	1
DFDF31	Masked T1	*	a	var
DFDF32	Masked T2 Status	*	b	1
DFDF33	Masked T2	*	a	var
DFDF34	Masked T3 Status	*	b	1
DFDF35	Masked T3	*	a	var
DFDF36	Encrypted T1 Status	*	b	1
DFDF37	Encrypted T1	*	b	var
DFDF38	Encrypted T2 Status	*	b	1
DFDF39	Encrypted T2	*	b	var
DFDF3A	Encrypted T3 Status	*	b	1
DFDF3B	Encrypted T3	*	b	var
DFDF3C	Encrypted MagnePrint	*	b	56
DFDF3D	MS2.0 Status	*	b	8
DFDF3F	CAPK Tag	*	b	var
DFDF40	Signature Required: 0x01 = Signature Required 0x80 = CBC-MAC checked in ARQC online response	*	b	1
DFDF41	PIN KSN	*	b	10

Tag	Description	Default (HEX)	Format	Length
DFDF42	PIN Encryption Type: 0xxx xxxx = Fixed key 1xxx xxxx = DUKPT key xx00 xxxx = TDES xx01 xxxx = AES xxxx xx00 = Data variant xxxx xx01 = PIN variant xxxx xx10 = MAC variant	*	b	1
DFDF43	MagnePrint Status Data	*	b	4
DFDF44	Encrypted PAN Data	*	b	var
DFDF4D	Masked ICC Track2 data	*	a	var
DFDF50	MSR KSN	*	b	10
DFDF51	MSR Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF52	Card Type Reported as clear text in ARQC with possible values 01 or 07. See <b>Appendix C Status and Message Table</b> .	-	b	1
DFDF53	Fallback Indication 0x00=No fallback or missing tag 0x81=MSR Fallback used 0x01=Technical Fallback used	*	b	1
DFDF54	MAC KSN	*	b	10
DFDF55	MAC Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF56	Encrypted Transaction Data KSN	*	b	10
DFDF57	Encrypted Transaction Data Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF58	Number of Bytes of Padding in F8	*	b	1
DFDF59	Encrypted Data Primitive	*	b	var
DFDF61	BIN Table Slot 1	00 00 00 00 00 00	b	6
DFDF62	BIN Table Slot 2	00 00 00 00 00 00	b	6
DFDF63	BIN Table Slot 3	00 00 00 00 00 00	b	6
DFDF64	BIN Table Slot 4	00 00 00 00 00 00	b	6
DFDF65	BIN Table Slot 5	00 00 00 00 00 00	b	6
DFDF66	BIN Table Slot 6	00 00 00 00 00 00	b	6
DFDF67	Acquirer Terminal Config - Fallback (0=Fallback Not Supported, 1=Fallback Supported)	01	b	1

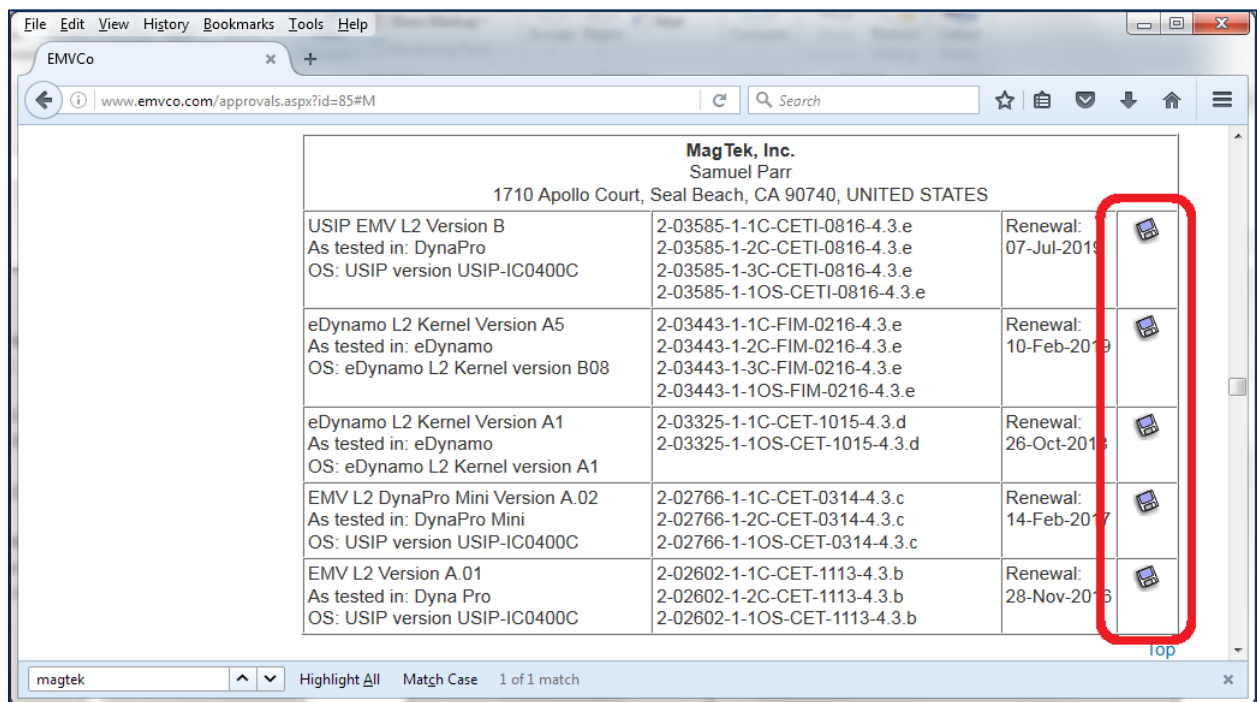
Tag	Description	Default (HEX)	Format	Length
DFDF68	Acquirer Terminal Config - PIN Bypass (0=PIN Bypass Not Supported, 1-PIN Bypass Supported)	01	b	1
DFDF70	Terminal Action Code - Default	00 00 00 00 00 00	b	5
DFDF71	Terminal Action Code - Denial	00 00 00 00 00 00	b	5
DFDF72	Terminal Action Code - Online	00 40 00 00 00 00	b	5
DFDF73	Payment Brand Account Type 0x00 = Default 0x01 = Credit, Debit, or Default 0x02 = Debit 0x03 = Credit	00	b	1

\* - Value is based on the ongoing transaction

## Appendix E EMV Configurations (EMV Only)

For the most up-to-date information about the device’s EMV Terminal Configuration, EMV Terminal Type, EMV Terminal Capabilities, and Additional EMV Terminal Capabilities, see the EMVCo Letter Of Approval (LOA) for the device:

- 1) In a web browser, open [www.emvco.com](http://www.emvco.com).
- 2) Follow the **Approvals and Certification** link.
- 3) Expand the navigation tree to **Terminal Type Approval** > **Approved Products**, and follow the **Level 2 Contact Approved Application Kernels** link.
- 4) Alternatively you may try [this direct table link to the M section of tables](#).
- 5) Find the table for products made by **MagTek, Inc.** and locate the table row for the device you are working with.
- 6) Click the attachment icon at the end of the row to open the Letter of Approval for that device.



## Appendix F Error Codes

### F.1 H Codes

Table 3-78 - "H" Error Codes

Code Displayed	Description	Comments
H1	Internal SYSCLK error	USIP bad
H2	Internal GPIO error	USIP bad
H3	Internal RNG error	USIP bad
H4	Internal RTC error	USIP or RTC crystal bad
H5	Internal Timer Init failed	USIP bad
H6	Crypto Engine Init failed	USIP bad
H7	Internal SRAM test failed	USIP bad
H8	Crypto Library Init failed	USIP bad or flash programming failed
H9	HAL Library Init failed	USIP bad
H10*	Crypto Library Self-Test failed	USIP bad or flash programming failed
H11*	Firmware Signature Check error	FW contents tampered with or failed
H15	Keypad Controller Init failed	Cirque controller or device SPI is bad
H16	Unable to communicate with MSR or mismatched key	MSR, SPI, or cable bad
H17	SDRAM, Touch Panel, Flash, K2 power-up failure	Peripheral or electronic switch bad
H18	iOS Security Chip Failure	I2C bad
H19	AMS 3911 Failure	Communications with AMS chip failed
H20	GPIO Extender Failure	GPIO Extender Circuit bad
H21	Battery Charge Measurement Failure	Battery Charge Measurement Circuit bad
H22	ADC Failure	USIP bad



## F.2 S Codes

Table 3-79 - "S" Error Codes

Code Calculated Sn where n = sum of the following	Description	Meaning
+1	Master key storage (BPK) erased or bad	Usually occurs after battery reset or battery power loss
+2	Keypad calibration not complete (Status stored in BPK)	After reset, calibration should be redone (but really only status has been lost)
+4	MSR key pairing not completed	Need to perform action. Permanent.
+8	Tamper sensors not activated	Status set after BPK initialized. Cleared once sensors activated. Stored in BPK
+16	Keypad activation sequence not sent	Need to perform action. Permanent.
+32	Keypad tamper detected during power up (Threshold stored in BPK)	Retry. Keypad may have shifted or been modified. Something nearby may be interfering. Keypad calibration bad or not performed yet.

Table 3-80 - Common "S" Error Codes

Common S codes	Description	Meaning
S63/S31	Usually new unit	Usually occurs after battery reset or battery power loss
S43/S35	Keypad calibration not complete (Status stored in BPK)	After reset, calibration should be redone (but really only status has been lost)

## F.3 C Codes

Table 3-81 - "C" Error Codes

Code Calculated Cn where n = sum of the following	Description
+1	Mfg Unbind certificate not loaded
+2	Device CA certificate not loaded
+4	PIN CA certificate not loaded

Code Calculated Cn where n = sum of the following	Description
+8	MSR CA certificate not loaded
+16	Device certificate does not exist

## F.4 Device Offline K Codes

Table 3-82 - "K" Error Codes

Code Displayed	Description
K15	No MSR keyload cert or key installed No PIN keyload cert or key installed
K13	No MSR keyload cert or key installed No PIN key installed (or exhausted)
K12	No MSR keyload cert or key installed
K7	No PIN keyload cert or key installed No MSR key installed (or exhausted)
K5	No MSR key installed (or exhausted) No PIN key installed (or exhausted)
K4	No MSR key installed (or exhausted)
K3	No PIN keyload cert or key installed
K1	No PIN key installed (or exhausted)

## F.5 Device offline A Codes

Table 3-83 - "A" Error Codes

Code Displayed	Description
A00	Device awaiting authentication

## Appendix G Factory Defaults

This section lists device configuration tags available in the device for use with the commands in section **3.6 EMV-Related Commands and Reports**.

### G.1 Certificate Authority Public Keys

Certificate Authority Public Key (CAPK) slots will be left blank.

### G.2 EMV Contact Factory Defaults (EMV Only)

Details about the tag set in this section are provided in *EMV Integrated Circuit Card Specifications for Payment Systems v4.3*.

Some tags are stored in a common contactless database; changing the value of the tag in one database will cause the tag to be changed in all databases. Common tags are 5F36, 9F1A, 9F1C, DFDF14, DFDF15, DFDF19, and 9F4E.

#### G.2.1 EMV Contact Terminal Factory Defaults

**Table 3-84 - EMV Contact Terminal Factory Defaults**

Tag Description	Tag	Configurable	Default Value (hex)
PSE Name	0xDFDF22	Compile Only	31 50 41 59 2E 53 59 53 2E 44 44 46 30 31
Number of Applications	0xDFDF21	Compile Only	0A
Transaction Currency Code	0x5F2A	MagTek	08 40
Transaction Currency Exponent	0x5F36	MagTek	02
Terminal Country Code	0x9F1A	MagTek	08 40
IFD Serial Number	0xDFDF25	Read Only	XX XX XX XX XX XX XX XX (USIP SN Binary)
Terminal ID	0x9F1C	MagTek	31 31 32 32 33 33 34 34 (Configurable)
Terminal Capabilities	0x9F33	Manufacturing	ICS Config 1:E0 F8 C8, ICS Config 2:E0 B8 C8
Terminal Type	0x9F35	Manufacturing	ICS Config 1:22, ICS Config 2:22
Additional Terminal Capabilities	0x9F40	Manufacturing	ICS Config 1:70 00 A0 B0 01, ICS Config 2:70 00 A0 B0 01
Random number for random online transaction selection	0xDFDF00	Device	8D
Revoked Certificate Lists	0xDFDF01	Not supported	A0 00 00 00 03 96 FF FF FF A0 00 00 00 04 96 FF FF FF A0 00 00 00 05 96 FF FF FF
Authorization Request Tags (ARQC)	0xDFDF02	MagTek	9F 03 9F 26 82 5A 5F 34 9F 36 9F 1A 95 9F 02 5F 2A 9A 9C 9F 37 9F 10 DF DF 53 F5 F4

Tag Description	Tag	Configurable	Default Value (hex)
Advice Tags	0xDFDF03	Not supported	5A
Financial Request Tags (ARPC)	0xDFDF04	Not supported	91 71 72 9F 01 89 8A
Reversal Tags	0xDFDF05	MagTek	82 9F 36 DF DF 25 9F 10 9F 5B 9F 33 9F 35 95 9F 01 5F 24 5A 5F 34 8A 9F 15 9F 16 9F 39 9F 1A 9F 1C 57 9F 02 5F 2A 9A 9F 21
Authorization Response Tags	0xDFDF06	MagTek	8A 91
Certification Validation Table	0xDFDF07	Not supported	00
Default CVM	0xDFDF13	MagTek	01
Socket Timeout	0xDFDF14	MagTek	00 00 0B B8
Socket Retries	0xDFDF15	MagTek	00 00 00 01
Issuer Script Max Size	0xDFDF16	Compile Only	00 00 00 80
Batch Data Tags	0xDFDF17	MagTek	82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 10 9F 26 9F 27 9F 36 95 9B 9C 9F 33 9F 34 9F 37 9F 40 DF DF 70 DF DF 71 DF DF 72 9F 5B
Default Terminal Language	0xDFDF19	MagTek	65 6E
Terminal Features	0xDFDF20	Manufacturing	57
Acquirer Terminal Config - Fallback	0xDFDF67	MagTek	01
Acquirer Terminal Config - PIN Bypass	0xDFDF68	MagTek	01
Terminal/Payment Brand Label	0xDFDF26	MagTek	00 00 00 ...00 [16 bytes fixed]
Terminal/Payment Brand Database Checksum	0xDFDF27	Device	Calculated - 20 Bytes
CAPK Label	0xDFDF28	MagTek	00 00 00 ...00 [16 bytes fixed]
CAPK Database Checksum	0xDFDF29	Device	Calculated - 20 Bytes
Supported Terminal Languages	0xDFDF2D	Manufacturing	65 6e 66 72 69 74 64 65 65 73

**G.2.2 EMV Contact Payment Brand Factory Defaults**

**Table 3-85 - EMV Contact Payment Brand Factory Defaults - Slot 0**

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 04 10 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 04 10 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Slots 1 through 9 are left empty.

## Appendix H Language and Country Codes

The device's language and country codes are derived from *ISO 3166-1*; country codes are numeric, and language codes are ASCII strings based on alpha-2.

### H.1 Terminal Country Codes

Table 3-86 - Terminal Country Codes

0840	United States
0250	France
0380	Italy
0724	Spain
0276	Germany

### H.2 Terminal Language Codes

Table 3-87 - Terminal Language Codes

656E	English (en)
6672	French (fr)
6974	Italian (it)
6465	German (de)
6573	Spanish (es)

## Appendix I Bluetooth LE Module Control Data (Bluetooth LE Only)

This section defines control messages that can be sent to the device's Bluetooth LE module. The general format for control messages is as follows, and the specific commands can be found in subsequent sections:

Byte 0            Message type  
                  0 = Request  
                  1 = Response  
                  2 = Notification

### Request message type

Byte 1            Command identifier  
Byte 2..n        Command request data

### Response message type

Byte 1            Response Code  
                  0 (success)  
                  1 (failure)  
                  2 (bad parameters)  
Byte 2..n        Command response data

### Notification message type

Byte 1            Notification identifier  
Byte 2..n        Notification data

## I.1 Bluetooth LE Module Configuration Properties

### I.1.1 Get Property Command

#### Request message

Byte 0            0 (Request message type)  
Byte 1            0 (Get property command identifier)  
Byte 2            Property identifier

#### Response message

Byte 0            1 (Response message type)  
Byte 1            Response code  
Byte 2..n        Property value

### I.1.2 Set Property Command

#### Request message

Byte 0            0 (Request message type)  
Byte 1            1 (Set property command identifier)  
Byte 2            Property identifier  
Byte 3..n        Property value

#### Response message

Byte 0            1 (Response message type)  
Byte 1            Response code

### I.1.3 Software ID Property

Property ID: 0x00

Get Property: Yes  
Set Property: No  
Default value: None

Description: This is the 11 byte read-only property that identifies the software part number and version for the device. The first 8 bytes represent the part number and the last 3 bytes represent the version. For example this string might be “30050884A01”. This string is subject to change.

Example Get Software ID property:  
Request message (hex): 00 00 00  
Response message (hex): 01 00 33 30 30 35 30 38 38 34 41 30 31

#### I.1.4 Bluetooth Device Address Property

Property ID: 0x01  
Get Property: Yes  
Set Property: No  
Default value: None

Description: This is a 6 byte read-only property that contains the Bluetooth device address. The first byte contains the least significant byte of the address. This address will vary with each device.

Example Get Bluetooth Device Address property:  
Request message (hex): 00 00 01  
Response message (hex): 01 00 EC 11 A0 E5 C5 78

#### I.1.5 Bluetooth Device Name Property

Property ID: 0x02  
Get Property: Yes  
Set Property: Yes  
Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset. This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: For DynaPro Mini, string “DynaProMini-XXYY”, where XX is the second to least significant byte of the Bluetooth device address converted to ASCII hex, and YY is the least significant byte. For example, if the second to least significant byte of the Bluetooth device address is 0x11 and the least significant byte is 0xEC, the Bluetooth device name would be “DynaProMini-11EC”. To reset the device to this default, set this property using a zero-length string. Shipped (factory default) values may differ. For example, some devices may be shipped with the last five characters of the Device Name property set to the last five characters of the device’s serial number.

Description: This property contains the Bluetooth device name, which is typically used by the software to present a cardholder or operator with a list of devices to interact with. To avoid ambiguity, if the solution specifies that more than one device of the same name will be available, MagTek recommends including a unique identifier in the device name so the cardholder or operator can differentiate.

The Bluetooth device name can have a length of 0 to 16 ASCII characters. The device will accept a device name up to 20 ASCII characters long, but the device will only use the first 16 characters when advertising over Bluetooth LE. Use caution when deciding whether to use more than 16 characters to avoid cardholder or operator confusion between the device name reported via Bluetooth LE and the device name reported via direct connections. The name should not contain any null string characters



(0x00). If set to a length of 0 the value will revert back to its original default value described above. After modifying the Bluetooth device name, you must reset the Bluetooth LE module. See section 2.3 **How to Use Bluetooth LE Connections** for details.

Example Get Bluetooth Device Name property:

Request message (hex): 00 00 02

Response message (hex): 01 00 31 32 33 (device name "123")

Example Set Bluetooth Device Name property:

Request message (hex): 00 01 02 31 32 33 (device name "123")

Response message (hex): 01 00

### I.1.6 Configuration Revision Property

Property ID: 0x03

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset. This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 0

Description: This property contains the configuration revision. The configuration revision can be used by the host to determine if the Bluetooth LE module's non-volatile properties have been configured and if so to what revision they have been configured to. For example, after the host configures the Bluetooth LE module's non-volatile properties to a certain state it may also want to bump this configuration revision to 1. Then the host can check this configuration revision on power up and use it to determine if the Bluetooth LE module's non-volatile properties have been configured to the desired state or not. For example if the configuration revision is 0 then the module needs to be configured and if it is set to 1 then it has already been configured to the desired configuration.

The Bluetooth LE module's non-volatile properties are stored in flash memory so they should not be changed too many times or the flash memory may wear out. This property is a one byte value that can be set to any value between 0 and 255.

Example Get Configuration Revision property:

Request message (hex): 00 00 03

Response message (hex): 01 00 01 (configuration revision 1)

Example Set Configuration Revision property:

Request message (hex): 00 01 03 01 (configuration revision 1)

Response message (hex): 01 00

### I.1.7 Power Timeout Property

Property ID: 0x04

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset. This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 60 seconds

Description: This property contains the power timeout. The power timeout is used to automatically power off the main logic board if the main board is powered up by the power on/off switch and a Bluetooth connection is not established. The main board is also powered up when a Bluetooth connection is established and it is also powered down when a Bluetooth connection is terminated. The host-controlled Bluetooth LE Power Configuration command also controls power to the main board.

This property is a two byte value and its units are seconds. The first byte is the most significant byte. Setting the value to 0 will disable the timeout.

Example Get Power Timeout property:

Request message (hex): 00 00 04

Response message (hex): 01 00 00 3C (power timeout 60 decimal)

Example Set Power Timeout property:

Request message (hex): 00 01 04 00 3C (power timeout 60 decimal)

Response message (hex): 01 00

### I.1.8 Power Control Property

Property ID: 0x05

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 0x03 (button controlled and Bluetooth Connection controlled)

Description: This property contains the power control bit that determine what controls power to the main board. The host-controlled Bluetooth LE Power Configuration command also controls power to the main board. This property is a one byte value that is defined as follows:

**Table 3-88 - Bluetooth LE Power Control Property**

Bit	7	6	5	4	3	2	1	0
Name	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Bluetooth connection	Button

Button:

0: The button does not control the device power.

1: The button does control the device power. Pressing and releasing the button when the device is off will power on the main logic board. Pressing and releasing the button when the device is on will power off the main logic board.

Bluetooth connection:

0: The Bluetooth Connection does not control the device power.

1: The Bluetooth Connection does control the device power. Establishing a Bluetooth connection when the device is off will power on the main logic board. Terminating a Bluetooth connection when the device is on will power off the main logic board.

Example Get Power Control property:

Request message (hex): 00 00 05

Response message (hex): 01 00 03 (button controlled and Bluetooth Connection controlled)

Example Set Power Control property:

Request message (hex): 00 01 05 03 (button controlled and Bluetooth Connection controlled)

Response message (hex): 01 00

### I.1.9 Advertising Control Property

Property ID: 0x06

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 0x00

Description: This property contains the advertising control bits, which control when the device advertises using Bluetooth Low Energy. This is a one-byte value defined as follows:

**Table 3-89 - Bluetooth LE Advertising Control Property**

Bit	7	6	5	4	3	2	1	0
Name	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Reserved set to 0	Never Advertise	Advertise after reset

Advertise after reset:

0 (default): The device will not advertise after Bluetooth LE reset until the cardholder or operator either presses and releases the power button or connects the device to USB power (see section **2.3 How to Use Bluetooth LE Connections**).

1: The device will advertise automatically after being reset. The cardholder or operator can't turn off advertising using the power button.

Never Advertise:

0 (default value): The device will advertise.

1: The device will never advertise. This mode may be useful for operators who only want to use the USB interface and don't want the device to advertise.

Example Get Advertise Control property:

Request message (hex): 00 00 06

Response message (hex): 01 00 00

Example Set Advertise Control property:

Request message (hex): 00 01 06 00

Response message (hex): 01 00

### I.1.10 Bluetooth LE Passkey Property

Property ID: 0x07

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 0 (000000)

Description: This property contains the Bluetooth passkey.

This property is a four byte hex value that represents a six digit decimal number between 000000 and 999999. The first hex byte is the least significant byte (LSB). Because the maximum decimal value allowed is 999999, which is equivalent to the four byte hex value 00 0F 42 3F, (3F 42 0F 00 in LSB order), the last byte of the four byte LSB first hex value will always be 00.

Example Get Property:

Request message (hex): 00 00 07

Response message (hex): 01 00 3F 42 0F 00 (passkey 999999 decimal)

Example Set Property:

Request message (hex): 00 01 07 3F 42 0F 00 (passkey 999999 decimal)

Response message (hex): 01 00

### I.1.11 Desired Bluetooth LE Minimum Connection Interval Property

Property ID: 0x08

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 10 (12.5 milliseconds)

This property is a two byte integer in least significant byte order that contains the value in 1.25 millisecond units.

Changes made to this property will not take effect until the device is reset or power cycled.

Description: This property contains the value of Interval Min sent to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST. See the core Bluetooth specification for more details. Only values between 6 and 3200 are valid.

Example Get Property:

Request message (hex): 00 00 08

Response message (hex): 01 00 0A 00 (10 (0x0A) (12.5 milliseconds))

Example Set Property:

Request message (hex): 00 01 08 0A 00 (10 (0x0A) (12.5 milliseconds))

Response message (hex): 01 00

### I.1.12 Desired Bluetooth LE Maximum Connection Interval Property

Property ID: 0x09

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 10 (12.5 milliseconds)

This property is a two byte integer in least significant byte order that contains the value in 1.25 millisecond units.

Changes made to this property will not take effect until the device is reset or power cycled.

Description: This property contains the value of Interval Max sent to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST. See the core Bluetooth specification for more details. Only values between 6 and 3200 are valid.

Example Get Property:

Request message (hex): 00 00 09

Response message (hex): 01 00 0A 00 (10 (0x0A) (12.5 milliseconds))

Example Set Property:

Request message (hex): 00 01 09 0A 00 (10 (0x0A) (12.5 milliseconds))

Response message (hex): 01 00

### I.1.13 Desired Bluetooth LE Slave Latency Property

Property ID: 0x0A

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 4

This property is a two byte integer in least significant byte order that contains the value.

Changes made to this property will not take effect until the device is reset or power cycled.

Description: This property contains the Slave Latency value the device will send to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST. See the core Bluetooth specification for more details. Only values between 0 and 499 are valid.

Example Get Property:

Request message (hex): 00 00 0A

Response message (hex): 01 00 04 00 (4)

Example Set Property:

Request message (hex): 00 01 0A 04 00 (4)

Response message (hex): 01 00

### I.1.14 Desired Supervision Timeout Property

Property ID: 0x0B

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset.

This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 500 (5000 milliseconds)

This property is a two byte integer in least significant byte order that contains the value in 10 millisecond units.

Changes made to this property will not take effect until the device is reset or power cycled.

Description: This property contains the value of Timeout Multiplier sent to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST. See the core Bluetooth specification for more details. Only values between 10 and 3200 are valid.

Example Get Property:

Request message (hex): 00 00 0B

Response message (hex): 01 00 F4 01 (500 (0x1F4) (5000 milliseconds))

Example Set Property:

Request message (hex): 00 01 0B F4 01 (500 (0x1F4) (5000 milliseconds))

Response message (hex): 01 00

### I.1.15 Connection Parameter Update Request Control Property

Property ID: 0x0C

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes. Changes made to this property will persist even if the device is powered off or reset. This property should only be changed once during device configuration. Modifying this property too many times will wear out flash memory.

Default value: 0x01 (send connection parameter update bit is set)

Description: This property is a one byte value that contains the connection parameter update control bits. The connection parameter update control bits can be used to control various connection parameter update features. The most significant bit of the byte is bit 7 and the least significant is bit 0. At this point bits 7-1 are reserved for future use and should always be set to 0.

Bit 0 is the send connection parameter update bit. When this bit is set the device will send a connection parameter update request once after each Bluetooth LE connection.

Example Get property:

Request message (hex): 00 00 0C

Response message (hex): 01 00 01 (send connection parameter update bit is set)

Example Set property:

Request message (hex): 00 01 0C 01 (send connection parameter update bit is set)

Response message (hex): 01 00

## I.2 Other Commands

### I.2.1 Echo Command

This command echo's the data received in the request message by transmitting the same data in the response message.

Request message

Byte 0            0 (Request message type)

Byte 1            2 (Echo command identifier)

Byte 2-n          Data to echo

Response message

Byte 0            1 (Response message type)  
Byte 1            Response code  
Byte 2-n          Data to echo from request message

Example Echo command:

Request message (hex): 00 02 01 02 03

Response message (hex): 01 00 01 02 03

### I.2.2 Reset Command

This command can be used to reset the Bluetooth LE module. The module will start resetting 2 seconds after it receives this command.

Request message

Byte 0            0 (Request message type)  
Byte 1            3 (Reset command identifier)

Response message

Byte 0            1 (Response message type)  
Byte 1            Response code

Example Reset command:

Request message (hex): 00 03

Response message (hex): 01 00

### I.2.3 Erase All Non-volatile Memory Command

This command erases all the Bluetooth LE module's non-volatile memory which returns it to its un-configured factory default state. This includes erasing all bonds. See the Erase all bonds command for more details. The secure codes are required to make sure a cardholder or operator does not accidentally send this command. This command will automatically reset the Bluetooth LE module 2 seconds after it completes.

Request message

Byte 0            0 (Request message type)  
Byte 1            4 (Command identifier)  
Byte 2            (hex) 55 (Secure code 1)  
Byte 3            (hex) AA (Secure code 2)

Response message

Byte 0            1 (Response message type)  
Byte 1            Response code

Example command:

Request message (hex): 00 04 55 AA

Response message (hex): 01 00

### I.2.4 Erase All Bonds Command

This command erases all the Bluetooth LE module's bonds. It will have to be re-paired with any Bluetooth LE host it wants to communicate with. Remove the device from all paired hosts prior to trying to re-pair the device. If any previously paired hosts are still in range of the device after issuing this command they may still try to connect with the device thus causing the device to stop advertising which will make it unable to re-pair until the device is removed from that host. The secure codes are required to

make sure an operator does not accidentally send this command. This command will automatically reset the Bluetooth LE module 2 seconds after it completes.

Request message

Byte 0            0 (Request message type)  
Byte 1            5 (Command identifier)  
Byte 2            (hex) 55 (Secure code 1)  
Byte 3            (hex) AA (Secure code 2)

Response message

Byte 0            1 (Response message type)  
Byte 1            Response code

Example command:

Request message (hex): 00 05 55 AA

Response message (hex): 01 00