

eDynamo

Secure Card Reader Authenticator Programmer's Manual (COMMANDS)



July 2021

Manual Part Number:
D998200115-20

REGISTERED TO ISO 9001:2015

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. SOME FEATURES AND FUNCTIONS MAY BE DOCUMENTED, BUT NOT AVAILABLE WITH THE CURRENT RELEASE OF THE PRODUCT. PLEASE CONTACT YOUR MAGTEK REPRESENTATIVE FOR QUESTIONS ABOUT SPECIFIC FEATURES AND FUNCTIONS AND WHEN THEY ARE SCHEDULED TO BECOME AVAILABLE.

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.
IntelliStripe® 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.
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.
UL™ and the UL logo are trademarks of UL LLC.

PCI Security Standards Council® is a registered trademark of the PCI Security Standards Council, LLC.
EMV® is a registered trademark in the U.S. and other countries and an unregistered trademark elsewhere. The EMV trademark is owned by EMVCo, LLC. The Contactless Indicator mark, consisting of four graduating arcs, is a trademark owned by and used with permission of EMVCo, LLC.

The *Bluetooth*® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by MagTek is under license.

Apple Pay®, iPhone®, iPod®, and Mac® are registered trademarks of Apple Inc., registered in the U.S. and other countries. App StoreSM is a service mark of Apple Inc., registered in the U.S. and other countries. iPad™ and iPad mini™ are trademarks 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.
Google Play™ store and Android™ platform are trademarks of Google Inc.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.

USB (Universal Serial Bus) Specification is Copyright © 1998 Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, NEC Corporation.

Keyboard Usage Definitions content is taken from Universal Serial Bus HID Usage Tables, Version 1.12, Section 10, Keyboard/Keypad Page (0x07) ©1996-2005 USB Implementers' Forum
Modifier Byte Definitions content is taken from Section 8.3 Report Format for Array Items, Device Class Definition for Human Interface Devices (HID) Version 1.11, ©1996-2001 USB Implementers' Forum, hidcomments@usb.org.

Some device icons courtesy of <https://icons8.com/>, used under the Creative Commons Attribution-NoDerivs 3.0 license.

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

Table 0-1 - Revisions

Rev Number	Date	Notes
10	January 2016	Initial Release derived from master programmer's manual
11	October 28, 2016	Miscellaneous clarifications and cleanup derived from master programmer's manual rev 11
15	May 30, 2018	Derived from master programmer's manual rev 15: Find instances of eDynamo and add mDynamo where applicable; Add MSR Only/Keypad Entry Only to properties and commands they pertain to; fix properties per device table for mDynamo; fix EMV slot 2 9F06 and 9F01 default settings; Add SL2 (not encrypting) option for DynaPro format EMV data; Promote former Appendix D to section 5; Add detail from <i>D99875433-41</i> about Command 0x14 - Get Device State (MSR Only) ; Add section 6.1 per request from sales/support; Add fixed key features; Improve clarity of encryption-related features; Add command security information for mDynamo; Replace cross-references to unpublished <i>Audio Reader Communication Protocol_v1</i> with in-place content; Change "manual entry" to "keypad entry" for clarity; Add missing example tables for extended commands in Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only) ; Add detail to Extended Command 0x0300 - Initiate EMV Transaction (EMV Only) ; Fix section 5 to accommodate EMV-only devices; Expose Property 0x67 - EMV Data Encryption Variant (EMV Only) , Property 0x54 - Card Data Encryption Variant (MSR Only, Configurable MSR Variants Only) , Property 0x56 - MagnePrint Data Encryption Variant (MSR Only, Configurable MagnePrint Variants Only) ; Move KSN interpretation info to Command 0x09 - Get Current TDES DUKPT KSN to provide details for devices that do not have EMV; Clarity rewrites of Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only) , Remaining MSR Transactions ; Add Dynasty, kDynamo, mDynamo Contactless Module, pDynamo, tDynamo; Remove vestigial Properties Per Device table from section 9 (now covered by section heading tags); Add Property 0x52 - Host Poll Timeout (HID Only KB Only) ; Add cDynamo, Dynamag Duo; Add Property 0x6D - EMV Contact Notification Configuration (Contact Only) ; Modify Property 0x33 - Card Inserted (MSR Insert Only Contact Only) ; Multiple clarifications to Masked Track Data and Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only) , including Service Code unmasked on eDynamo and mDynamo; Change "insert a card" "swipe a card" and "tap a card" to "present payment" (reviewed by EL and HM); Tag relevant features and values as Only Contact / Online; Retrofit Unattended Operation feature,

Rev Number	Date	Notes
		<p>notably in Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only); Update device defaults in Property 0x10 - Interface Type; Add tDynamo, kDynamo, mDynamo Contactless; Major rewrite to streamline and add to Table 1-2; Add options in Extended Command 0x030B - Read EMV Kernel Information to retrieve contactless kernel version, checksum, and configuration info; Add Appendix E EMV Terminal and Application Settings (EMV Only); Clarify forming DSN in Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only); Correct example in Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only); Compare, reconcile, merge contents of <i>D99875483-6.01</i>, missed during original master merge, including data types in section 6; Retrofit filtering out of properties by SureSwipe feature; Update contactless statuses in Notification 0x0300 - Transaction Status / Progress Information; Large update to Table 1-2 - Device Features; Update examples in Extended Command 0x0305 - Modify Terminal Configuration (MAC), Extended Command 0x0307 - Modify Application Configuration (MAC); Add Quick Chip option in Extended Command 0x0300 - Initiate EMV Transaction (EMV Only); Add tag 9F40 in Extended Command 0x0311 - Read EMV Configuration (Contact Only); Update available configurations in Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only), Extended Command 0x0311 - Read EMV Configuration (Contact Only); Extract tags common to all EMV databases into E.1 EMV Common Settings; Major clarity rewrite of introduction to EMV transactions in section 8.4; Remove legacy “original format” for EMV messages and merge EMV message formats under a single Appendix D; Separate iAP1 from iAP2 connection type; Misc. clarifications and corrections</p>
17	Feb 8, 2019	<p>Derived from master programmer’s manual rev 17: Throughout, clarify ANS specification and key names/variants; Add clarifying cross-references between sections about masked and encrypted card data; Section 7.2.1 add code 0x29; Section 1.5 update JIS Capable feature for SPI Encrypting IntelliHead V5 and UART Enc IntelliHead V5; Add JIS Capable and Set Mask Service Code features to Dynamag and USB Enc IntelliHead V5; Add Quick Chip feature and more supporting information; Remove deprecated products including Flash reader, Home Banking (Dynamo LCD), iDynamo throughout; Remove deprecated connection types Proprietary Wireless and 30-pin throughout; Remove deprecated features Store and Forward,</p>

Rev Number	Date	Notes
		<p>Custom Messages (redundant with Display),.and strip out properties no longer needed; Change feature Unattended Mode to OEM Features; Rotate feature table for space constraints; For clarity, remove redundant feature tags from section headings that implicitly inherit from parent sections; Section 4.3 add “C” as a security option; Throughout, clarify Security Level 2 behavior and impact on MagnePrint values and security; Misc. clarifications and corrections.</p>
20	Jul 7, 2021	<p>Derived from master programmer’s manual rev 20: Add Contactless Quick Chip feature and supporting detail, split EMV Quick Chip feature into Contact vs. Contactless; Add OEM Features feature; Update Table 1-2 and add iDynamo 5 (Gen II), iDynamo 6 and Power Management scheme PM7; Add External PIN Accessory Support feature; Add Application Selection Options feature; Add Dual USB Ports as an explicit feature; In Notification 0x0300 - Transaction Status / Progress Information add Progress Indicator 0x91 and others, tweak descriptions for clearer use in major clarity and completeness updates in section 8.4.2 About EMV L2 Transaction Flows (EMV Only); Add QuickPass Support feature and supporting information; Remove Dynasty; Refactor section 2.1.3 to improve severability between input report info for MSR vs. Notifications; Move terminal country codes and terminal language codes into tag tables instead of a dedicated appendix; Throughout, clarify Currency Codes are driven by ISO standard; 1.5 add features Pairing Mode Control, Apple VAS, Conserve DUKPT Keys, No EMV MSR Flow; Extended Command 0x0300 - Initiate EMV Transaction (EMV Only) add Apple VAS options and add CAUTION statements about device power and key consumption; Add Property 0x74 - EMV Transaction Result Format (EMV Only, Conserve DUKPT Keys Only); Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only) clarify parameter value vs. LoA value and the nature of the full list vs. LoA list, remove “in some LoAs this is referred to as” text to reflect LoAs being harmonized across product family, add configurations C8 through C13; Appendix D.1.1 label values of DFDF53 that are EMV MSR Flow Only; Misc. clarifications and corrections.</p>

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.....	7
FCC WARNING STATEMENT.....	8
FCC COMPLIANCE STATEMENT.....	8
CANADIAN DOC STATEMENT.....	8
CE STANDARDS.....	8
UL/CSA	8
RoHS STATEMENT	8
Table of Contents.....	9
1 Introduction	15
1.1 About This Document	15
1.2 About SDKs	15
1.3 About Terminology	16
1.4 About Connections and Data Formats.....	17
1.5 About Device Features	19
2 Connection Types.....	23
2.1 How to Use USB Connections (USB Only)	23
2.1.1 About USB Reports, Usages, Usage Pages, and Usage IDs.....	24
2.1.2 How to Send Commands On the USB Connection.....	25
2.1.3 How to Receive Data On the USB Connection (HID Only).....	27
2.2 How to Use Bluetooth LE Connections (Bluetooth LE Only).....	29
2.2.1 About GATT Characteristics.....	29
2.2.2 How to Connect to a Device Using Bluetooth LE.....	31
2.2.3 How to Send Commands On the Bluetooth LE Connection.....	31
2.2.4 How to Receive Data On the Bluetooth LE Connection.....	32
3 Data Formats	33
3.1 How to Use HID Format (HID Only)	33
3.2 How to Use GATT Format (GATT Only).....	34
4 Security Levels	36
4.1 About Message Authentication Codes (MAC).....	36
4.2 Security Level 2	36
4.3 Security Level 3	36
4.4 Security Level 4 (MSR Only).....	37
4.5 Command Behaviors By Security Level.....	37
5 Encryption, Decryption, and Key Management.....	39
5.1 About Encryption and Decryption	39
5.2 How to Determine the Key.....	39
5.3 How to Decrypt Data.....	40
6 Magnetic Stripe Card Data Sent from Device to Host (MSR Only Keypad Entry Only)	41

6.1	About Track Data.....	43
6.2	Track 1 Decode Status (HID TLV GATT SLIP).....	44
6.3	Track 2 Decode Status (HID TLV GATT SLIP).....	44
6.4	Track 3 Decode Status (HID TLV GATT SLIP, 3-Track Only).....	44
6.5	Card Encode Type (HID TLV GATT SLIP).....	45
6.6	Device Encryption Status	46
6.7	Encrypted Track Data	47
6.7.1	Track 1 Encrypted Data Length (HID GATT SLIP).....	47
6.7.2	Track 2 Encrypted Data Length (HID GATT SLIP).....	47
6.7.3	Track 3 Encrypted Data Length (HID GATT SLIP, 3-Track Only).....	48
6.7.4	Track 1 Absolute Data Length (HID GATT SLIP).....	48
6.7.5	Track 2 Absolute Data Length (HID GATT SLIP).....	48
6.7.6	Track 3 Absolute Data Length (HID GATT SLIP, 3-Track Only).....	49
6.7.7	Track 1 Encrypted Data.....	49
6.7.8	Track 2 Encrypted Data.....	49
6.7.9	Track 3 Encrypted Data.....	49
6.8	MagnePrint Status	51
6.9	MagnePrint Data Length (HID GATT SLIP)	53
6.10	MagnePrint Absolute Data Length (HID TLV GATT SLIP).....	53
6.11	Encrypted MagnePrint Data	53
6.12	Device Serial Number.....	54
6.13	Masked Track Data.....	55
6.13.1	About Masking.....	55
6.13.2	Track 1 Masked Data Length (HID GATT SLIP)	55
6.13.3	Track 2 Masked Data Length (HID GATT SLIP)	56
6.13.4	Track 3 Masked Data Length (HID GATT SLIP, 3-Track Only).....	56
6.13.5	Track 1 Masked Data	56
6.13.6	Track 2 Masked Data	57
6.13.7	Track 3 Masked Data (3-Track Only).....	58
6.14	Encrypted Session ID.....	58
6.15	DUKPT Key Serial Number (KSN).....	58
6.16	Remaining MSR Transactions.....	59
6.17	MagneSafe Version Number (HID GATT SLIP).....	59
6.18	SHA-1 Hashed Track 2 Data (HID TLV GATT SLIP, SHA-1 Only).....	59
6.19	HID Report Version (HID GATT SLIP).....	60
6.20	MagnePrint KSN (HID TLV GATT SLIP).....	60
6.21	Battery Level (HID GATT SLIP).....	60
7	Notification Messages Sent from Device to Host (Extended Notifications Only).....	62
7.1	About Notification Messages.....	62
7.2	Notification Group 0x03 - EMV L2 (EMV Only)	64

7.2.1	Notification 0x0300 - Transaction Status / Progress Information	64
7.2.2	Notification 0x0301 - Display Message Request.....	66
7.2.3	Notification 0x0302 - Cardholder Selection Request (EMV Only)	67
7.2.4	Notification 0x0303 - ARQC Message	68
7.2.5	Notification 0x0304 - Transaction Result Message	69
8	Commands	70
8.1	About Commands	70
8.2	About Result Codes.....	71
8.3	General Commands	72
8.3.1	Command 0x00 - Get Property	72
8.3.2	Command 0x01 - Set Property (MAC).....	73
8.3.3	Command 0x02 - Reset Device (MAC)	74
8.3.4	Command 0x09 - Get Current TDES DUKPT KSN	75
8.3.5	Command 0x0A - Set Session ID (MSR Only).....	76
8.3.6	Command 0x10 - Activate Authenticated Mode (MSR Only).....	77
8.3.7	Command 0x11 - Activation Challenge Response (MSR Only).....	79
8.3.8	Command 0x12 - Deactivate Authenticated Mode (MSR Only)	80
8.3.9	Command 0x14 - Get Device State (MSR Only).....	81
8.3.10	Command 0x15 - Get / Set Security Level (MAC).....	84
8.3.11	Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only).....	85
8.3.12	Command 0x45 - Get Battery Percentage (PM3 Only PM4 Only PM5 Only PM6 Only PM7 Only).....	86
8.3.13	Command 0x46 - Send Command to Bluetooth LE Controller (Bluetooth LE Only)....	87
8.3.13.1	Bluetooth LE Command 0x00 - Get Property	88
8.3.13.2	Bluetooth LE Command 0x01 - Set Property	88
8.3.13.3	Bluetooth LE Command 0x02 - Echo.....	88
8.3.13.4	Bluetooth LE Command 0x06 - Erase All Non-Volatile Memory.....	89
8.3.13.5	Bluetooth LE Command 0x07 - Erase All Bonds.....	90
8.3.13.6	Bluetooth LE Command 0x0B - Terminate Bluetooth LE Connection	91
8.3.13.7	Bluetooth LE Command 0x0D - Get Bond Count (Pairing Modes Only)	91
8.3.14	Command 0x48 - Notification Output Connection Override (Bluetooth LE Only iAP Only, USB Only).....	92
8.3.15	Command 0x49 - Send Extended Command Packet (Extended Commands Only)	93
8.3.16	Command 0x4A - Get Extended Response (Extended Commands Only).....	95
8.3.17	Command 0x4C - Get Tamper Status (Tamper Only).....	96
8.3.18	Command 0x4D - Configure General Status LED (PM3 Only).....	98
8.4	Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)	99
8.4.1	About MACs.....	99

8.4.2	About EMV L2 Transaction Flows (EMV Only).....	99
8.4.3	Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)	104
8.4.4	Extended Command 0x0302 - Cardholder Selection Result	109
8.4.5	Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)	111
8.4.6	Extended Command 0x0304 - Cancel Transaction (EMV Only).....	113
8.4.7	Extended Command 0x0305 - Modify Terminal Configuration (MAC)	114
8.4.8	Extended Command 0x0306 - Read Terminal Configuration.....	116
8.4.9	Extended Command 0x0307 - Modify Application Configuration (MAC).....	118
8.4.10	Extended Command 0x0308 - Read Application Configuration	120
8.4.11	Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)	122
8.4.12	Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only).....	125
8.4.13	Extended Command 0x030B - Read EMV Kernel Information	127
8.4.14	Extended Command 0x030C - Set Date and Time (MAC).....	129
8.4.15	Extended Command 0x030D - Read Date and Time	131
8.4.16	Extended Command 0x030E - Commit Configuration.....	133
8.4.17	Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only).....	135
8.4.18	Extended Command 0x0311 - Read EMV Configuration (Contact Only)	141
9	Properties.....	143
9.1	About Properties.....	143
9.2	Property 0x00 - Firmware ID	143
9.3	Property 0x01 - USB Serial Number (HID Only KB Only)	144
9.4	Property 0x02 - USB Polling Interval (HID Only KB Only)	145
9.5	Property 0x03 - Device Serial Number.....	146
9.6	Property 0x04 - MagneSafe Version Number	147
9.7	Property 0x05 - Track ID Enable (MSR Only).....	148
9.8	Property 0x07 - ISO Track Mask	149
9.9	Property 0x08 - AAMVA Track Mask (MSR Only).....	150
9.10	Property 0x0A - USB HID Max Packet Size (HID Only).....	151
9.11	Property 0x10 - Interface Type.....	152
9.12	Property 0x15 - MagnePrint Flags (MSR Only)	153
9.13	Property 0x31 - Mask Other Cards (MSR Only).....	154
9.14	Property 0x33 - Card Inserted (MSR Insert Only Contact Only)	155
9.15	Property 0x34 - Send AAMVA Card Data Unmasked (MSR Only).....	156
9.16	Property 0x38 - HID SureSwipe Flag (SureSwipe Only, HID Only, MSR Only).....	157
9.17	Property 0x52 - Host Poll Timeout (HID Only KB Only).....	158
9.18	Property 0x54 - Card Data Encryption Variant (MSR Only, Configurable MSR Variants Only)	159
9.19	Property 0x56 - MagnePrint Data Encryption Variant (MSR Only, Configurable MagnePrint Variants Only).....	160

9.20	Property 0x57 - SHA Hash Configuration (HID Only TLV Only, Configurable SHA Only, MSR Only).....	161
9.21	Property 0x5F - Notification Output Connection (Bluetooth LE Only iAP Only, USB Only) 162	
9.22	Property 0x67 - EMV Data Encryption Variant (EMV Only).....	163
9.23	Property 0x6D - EMV Contact Notification Configuration (Contact Only)	164
9.24	Property 0x74 - EMV Transaction Result Format (EMV Only, Conserve DUKPT Keys Only) 166	
Appendix A Bluetooth LE Controller Properties (Bluetooth LE Only).....		167
A.1	Bluetooth LE Property 0x00 - Bluetooth LE Firmware ID	167
A.2	Bluetooth LE Property 0x01 - Bluetooth LE Device Address	167
A.3	Bluetooth LE Property 0x02 - Bluetooth LE Device Name	168
A.4	Bluetooth LE Property 0x03 - Configuration Revision	169
A.5	Bluetooth LE Property 0x07 - Passkey.....	170
A.6	Bluetooth LE Property 0x08 - Configuration Bits.....	171
A.7	Bluetooth LE Property 0x0B - General Connection Timeout.....	173
A.8	Bluetooth LE Property 0x0C - Desired Minimum Connection Interval	174
A.9	Bluetooth LE Property 0x0D - Desired Maximum Connection Interval	175
A.10	Bluetooth LE Property 0x0E - Desired Slave Latency	176
A.11	Bluetooth LE Property 0x0F - Desired Supervision Timeout.....	177
A.12	Bluetooth LE Property 0x12 - Connection Parameter Update Request Control.....	178
A.13	Bluetooth LE Property 0x13 - Bluetooth Status LED Functionality Control (Pairing Modes Only) 179	
A.14	Bluetooth LE Property 0x15 - Pairable Timeout (Pairing Modes Only).....	180
A.15	Bluetooth LE Property 0x16 - Maximum Bond Count (Pairing Modes Only).....	181
A.16	Bluetooth LE Property 0x17 - Maximum Bond Mode (Pairing Modes Only)	182
A.17	Bluetooth LE Property 0x18 - Minimum Background Advertising Interval (Custom Advertising Only)	183
A.18	Bluetooth LE Property 0x19 - Maximum Background Advertising Interval (Custom Advertising Only)	184
A.19	Bluetooth LE Property 0x1C - Minimum Initial Advertising Interval (Custom Advertising Only) 185	
A.20	Bluetooth LE Property 0x1D - Maximum Initial Advertising Interval (Custom Advertising Only) 186	
A.21	Bluetooth LE Property 0x1E - Minimum Pairable Advertising Interval (Custom Advertising Only)	187
A.22	Bluetooth LE Property 0x1F - Maximum Pairable Advertising Interval (Custom Advertising Only)	188
Appendix B Examples.....		189
B.1	Command Examples.....	189
B.1.1	Example: HID Device Card Swipe In Security Level 2 (HID Only, MSR Only)	190
B.1.2	Example: Swipe Decryption, HID Device In Security Level 3 or 4 (HID Only, MSR Only) 194	
B.1.3	Example: Configuring a Device Before Encryption Is Enabled (HID Only)	203

B.1.4	Example: Changing from Security Level 2 to Security Level 3	204
B.1.5	Example: Changing from Security Level 2 to Security Level 4 (MSR Only).....	206
B.1.6	Example: Changing from Security Level 3 to Security Level 4 (MSR Only).....	208
B.1.7	Example: Authentication (MSR Only).....	210
B.2	About the SDKs and Additional Examples	213
Appendix C	Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only).....	214
C.1	ISO/ABA Financial Card	214
C.2	AAMVA Driver's License	215
Appendix D	EMV Message Formats (EMV Only)	216
D.1	ARQC Messages (EMV Only)	216
D.1.1	ARQC Message Format Security Level 2.....	216
D.1.2	ARQC Message Format Security Level 3.....	218
D.2	ARPC Response from Online Processing (EMV Only)	220
D.3	Transaction Result Messages (EMV Only)	221
D.3.1	Transaction Result Message Format Security Level 2	222
D.3.2	Transaction Result Message Format Security Level 3	223
Appendix E	EMV Terminal and Application Settings (EMV Only).....	225
E.1	EMV Common Settings.....	225
E.1.1	EMV Common Terminal Settings and Defaults.....	225
E.1.2	EMV Common Application Settings and Defaults.....	225
E.2	EMV Contact Settings (Contact Only)	226
E.2.1	EMV Contact Terminal Settings and Defaults (Contact Only)	226
E.2.2	EMV Contact Application Settings and Defaults (Contact Only)	230

1 Introduction

1.1 About This Document

This document describes how to communicate with Secure Card Reader Authenticator (SCRA) devices which implement MagneSafe V5.

1.2 About SDKs

MagTek provides convenient SDKs and corresponding documentation for many programming languages and operating systems. The API libraries included in the SDKs wrap the details of the connection in an interface that conceptually parallels the device's internal operation, freeing software developers to focus on the business logic, without having to deal with the complexities of platform APIs for connecting to the various available connection types, communicating using the various available protocols, and parsing the various available data formats. Information about using MagTek wrapper APIs is available in separate documentation, including *D99875535 Secure Card Reader Authenticator API PROGRAMMING REFERENCE MANUAL*.

The SDKs and corresponding documentation include:

- Functions for sending the direct commands described in this manual
- Wrappers for commonly used commands that further simplify development
- Sample source code to demonstrate how to communicate with the device using the direct commands described in this manual

To download the SDKs and documentation, search www.magtek.com for “SDK” and select the SDK and documentation for the programming languages and platforms you need, or contact MagTek Support Services for assistance.

Software developers also have the option to revert to direct communication with the device using libraries available in the chosen development framework. For example, custom software written in Visual Basic or visual C++ may make API calls to the standard Windows USB HID driver. This document provides information and support for developing host software using that method.

MagTek has also developed software that demonstrates direct communication with the device, which software developers can use to test the device and to which provides a starting point for developing other software. For more information, see the MagTek web site, or contact your reseller or MagTek Support Services.

1.3 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 Secure Card Reader Authenticator (SCRA) that receives and responds to the command set specified in this document. Devices include Dynamag, eDynamo, 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**.

1.4 About Connections and Data Formats

MagneSafe V5 products transmit data using a set of common data formats across a variety of physical connection layers, which can include universal serial bus (USB) acting as a keyboard (“USB KB”), USB acting as a vendor-defined HID device (“USB HID”), RS-232, Apple iAP (Lightning or USB), bidirectional audio connectors, Bluetooth, Bluetooth LE, and so on. The set of available physical connection types and the data formats available on each connection type is device-dependent. **Table 1-1** shows the physical connection types available on each product, and the data formats supported on each connection type for that device. Details about connection types and formats can be found in section **2 Connection Types** and section **3 Data Formats**. Section headings in this document include tags that indicate which connection types and/or data formats they apply to.

Table 1-1 - Device Connection Types / Data Formats

Product / Connection	Audio	Bluetooth LE GATT	Bluetooth LE GATT KB	Bluetooth	iAP1 Lightning	iAP2 Lightning	iAP2 USB	RS-232 / UART	SPI	USB HID	USB KB
BulleT KB				Streaming (MSR data)						HID	
BulleT SPP				Streaming							
cDynamo					Streaming						
Dynamag, Dynamag Duo, USB Enc IntelliHead V5										HID	Streaming
DynaMAX		GATT	Streaming							HID	
DynaPAD										HID	Streaming
DynaWave								SLIP		HID	
eDynamo		GATT								HID	
iDynamo 5					Streaming						
iDynamo 5 (Gen II)						Streaming					
iDynamo 6						SLIP	SLIP			HID	
kDynamo						SLIP					

Product / Connection	Audio	Bluetooth LE GATT	Bluetooth LE GATT KB	Bluetooth	iAP1 Lightning	iAP2 Lightning	iAP2 USB	RS-232 / UART	SPI	USB HID	USB KB
mDynamo										HID	
P-series and I-65 w/V5										HID	Streaming
pDynamo		GATT								HID	
sDynamo						Streaming					
SPI Enc IntelliHead V5									Streaming		
tDynamo		GATT								HID	
UART Enc IntelliHead V5								Streaming			
uDynamo	TLV									HID	

1.5 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 connection types, data formats, features, and configuration options, which affect the availability and behavior of some commands. **Table 1-2** provides a list of device features that may impact command availability and behavior. All section headings in this document include tags that indicate which features they apply to.

Table 1-2 - Device Features

Feature / Product	BulleT KB BulleT SPP	cDynamo	Dynamag, USB Enc IntelliHead V5	Dynamag Duo	DynaMAX	DynaPAD	DynaWave	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	mDynamo	P-series, I-65 w/V5	pDynamo	sDynamo	SPI Encrypting IntelliHead V5	tDynamo	UART Enc IntelliHead V5	uDynamo
MSR Swipe	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y
MSR Insert	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
MSR 3 Tracks	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	
MSR Disable	N	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
MSR Swap Tracks 1/3	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MSR Embedded V5 Head	N	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	N	Y	N	N
MSR Configurable MSR Variants		Y	Y	Y	Y		N	Y	Y	Y	Y	Y	Y		Y	Y		Y		
MSR Configurable MP Variants		N	N	N	Y		N	Y	N	N	Y	Y	Y		Y	N		Y		
MSR SureSwipe		N	Y	Y	Y	Y	N	Y	N	N	N	N	N	Y	N	N	N	N	N	N
MSR JIS Capable		Y	Y ³	N	N	N	N	N	Y	N	N	N	N	N	N	Y	Y	N	Y	
MSR SHA-1		N	Y	Y	Y	Y	N	Y	N	N	N	N	N		Y	N		N		
MSR SHA-256		N	N	N	N	N	N	N	N	N	N	N	N	N	N	N		N		
MSR Configurable SHA		N	N	N	Y		N	Y	N	N	N	N	N			N		N		
MSR MagneSafe 2.0							N	Y		N	N	N			N			N		
Configurable Encryption Algorithm	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N		N	N	N
Set Mask Service Code	N	N	Y ²	N	N	N	Y	N	N	N	N	N	N	Y ²	N	N	Y ²	N	N	N
Never Mask Service Code			N ²				N	Y	Y	Y	Y	Y	Y	N ²		Y	N ²	Y		

Feature / Product	BulleT KB	BulleT SPP	cDynamo	Dynamag, USB Enc IntelliHead V5	Dynamag Duo	DynaMAX	DynaPAD	DynaWave	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	mDynamo	P-series, I-65 w/V5	pDynamo	sDynamo	SPI Encrypting IntelliHead V5	tDynamo	UART Enc IntelliHead V5	uDynamo
EMV Contact	N	N	N	N	N	N	N	N	Y	N	N	Y	Y	Y	N	N	N	N	Y	N	N
EMV Contactless	N	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	Y	N	N
EMV ODA	N	N	N	N	N	N	N	Y	Y	N	N	N	Y	Y	N	N	N	N	Y	N	N
EMV MSR Flow	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	Y	N	N
No EMV MSR Flow	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	N	Y	Y
EMV Contact Quick Chip	N	N	N	N	N	N	N	N	Y ⁴	N	N	Y	Y	Y ⁴	N	N		N	Y	N	N
EMV Contactless Quick Chip	N	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	Y	N	N
Comprehensive Checksums	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Conserve DUKPT Keys	N	N	N	N	N	N	N	N	Y ⁷	N	N	N	N	Y ⁷	N	N	N	N	N	N	N
QuickPass Support	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Apple VAS	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Application Selection Options	N	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	Y	N	N
External PIN Accessory Support	N	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N
Keypad Entry	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Fixed Key	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
MSR Secondary DUKPT Key	N	N	N	N	Y	N	Y	Y	Y	N	N	N	N	Y	N	Y	N	N	N	N	Y
Power Mgt Scheme (PM#)	1	N	N	N	2	N	N	3	N	N	7	5	N	N	6	N	N	N	5	N	4
Battery-Backed RTC							N	Y		N	N	N	N					N			
OEM Features	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Transaction Validation	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N
Display	N	N	N	N	N	N	Y*	N	N	N	N	N	N	N	N	Y	N	N	N	N	N
Multi-Language	N	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	Y	N	N
Tamper	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N

Feature / Product	BulleT KB	BulleT SPP	cDynamo	Dynamag, USB Enc IntelliHead V5	Dynamag Duo	DynaMAX	DynaPAD	DynaWave	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	mDynamo	P-series, I-65 w/V5	pDynamo	sDynamo	SPI Encrypting IntelliHead V5	tDynamo	UART Enc IntelliHead V5	uDynamo
Extended Commands	N	N	N	N	N	N	N	Y	Y	N	N	Y	Y	Y	N	N	N	N	Y	N	N
Extended Notifications	N	N	N	N	N	N	N	Y	Y	N	N	Y	Y	Y	N	N	N	N	Y	N	N
Dual USB Ports	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	Y	N	N
Pairing Modes	N	N	N	N	N	N	N	N	Y ⁵	N	N	N	N	N	N	Y	N	N	Y	N	N
Pairing Mode Control	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N
Custom Advertising	N	N	N	N	N	N	N	N	Y ⁶	N	N	N	N	N	N	Y	N	N	Y	N	N
Configurable iAP FID	N	Y	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	N
Auxiliary Ports	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N
Configurable Baud Rate	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
Configurable Pushbutton	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	Y	N	N
External LED Control	N	N	N	N	N	N	N	N	N	N	N	N	Y ¹	N	N	N	N	N	N	N	N
Encrypt Bulk Data (b)	120	120	24	24	24	N	N	24	120	N	N	N	24	N	N	N	120	N	12	24	

Feature / Product	BulleT KB	BulleT SPP	cDynamo	Dynamag, USB	Enc IntelliHead V5	Dynamag Duo	DynaMAX	DynaPAD	DynaWave	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	mDynamo	P-series, I-65 w/V5	pDynamo	sDynamo	SPI Encrypting IntelliHead V5	tDynamo	UART Enc IntelliHead V5	uDynamo
<ol style="list-style-type: none"> 1) This feature is available in mDynamo firmware revision 1000003358D00 (released August 2017) and newer. 2) This feature was introduced in SPI Encrypting IntelliHead V5 in firmware version 21042876C01 released July 2017, P-series and I-65 w/V5 in firmware version 21165822E01 released March 2018, Dynamag and USB Encrypting IntelliHead V5 in firmware version 21042840K00 released January 2019. 3) This feature is available in Dynamag and USB Enc IntelliHead V5 firmware version 21042840K00 (released January 2019) and newer. 4) EMV Contact Quick Chip is available in mDynamo firmware revision 1000003358F01 (released December 2017), eDynamo firmware revision 1000003354F00 (released October 2018), and newer. 5) Pairing Modes feature is available in eDynamo firmware revision 1000002650B01 and newer, except Bluetooth LE Property 0x13 which was added in 1000002650C01. 6) Custom Advertising feature is available in eDynamo firmware revision 1000002650C02 and newer, except Bluetooth LE Property 0x08 Configuration Bits “Never Advertise” and “USB Power Not Exit Airplane Mode” which were added in 1000002650C01. 7) This feature is only supported by the eDynamo starting with firmware revision 1000003354J00 and mDynamo starting with firmware revision 1000003358G01. 																						

2 Connection Types

Table 1-1 on page **17** 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)

These USB devices conform to the USB specification revision 1.1. They also conform to the Human Interface Device (HID) class specification version 1.1. This document assumes the reader is familiar with USB HID class specifications, which are available at www.usb.org. MagTek strongly recommends becoming familiar with that standard before trying to communicate with the device directly via USB.

These devices are full-speed, high-powered USB devices that draw power from the USB bus they are connected to. They enter and wake up from Suspend mode when directed to do so by the USB host. They do not support remote wakeup.

When connecting via USB, MagneSafe V5 devices connect to the USB host either as a vendor-defined HID device (“HID”) or as an HID Keyboard Emulation device (“KB”), depending on the device type and configuration. Details for using the device in each of these modes are provided in the sections that follow. In addition to connecting to the USB host as different USB device types depending on their mode, the device can transmit data in different formats (see section **3 Data Formats**). To decode data coming from HID devices, see section **3.1 How to Use HID Format (HID Only)**.

The devices have an adjustable endpoint descriptor polling interval value that can be set to any value in the range of 1ms to 255ms. To change the setting, use **Property 0x02 - USB Polling Interval (HID Only | KB Only)**.

MagneSafe V5 devices identify themselves to the host with MagTek’s vendor ID **0x0801** and a Product ID (PID) from this list:

- MSR Swipe devices report PID **0x0011** when in HID mode.
- MSR Insert devices report PID **0x0013** when in HID mode.
- Audio devices report PID **0x0017** when in HID mode.
- Devices that implement a combination of EMV Contact / EMV Contactless / MSR swipe report PID **0x0019** when in HID mode.
- EMV-only devices (such as mDynamo and DynaWave) report PID **0x001A** when in HID mode.
- All devices report PID **0x0001** when in KB mode.
- Wireless USB device dongles report PID **0x0011** when in HID mode.
- Wireless USB device dongles report PID **0x0001** when in KB mode.
- Wireless USB devices report PID **0x0014** when plugged directly into the host with a USB cable.

2.1.1 About USB Reports, Usages, Usage Pages, and Usage IDs

All 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 (32-bit) identifier. The two most significant bytes of a usage are called the **usage page**, and the two least significant bytes are called the **usage ID**. Vendor-defined HID usages must have a usage page in the range **0xFF00 - 0xFFFF**, and it is common practice for related usage IDs share the same usage page. For these reasons, all usages for MagneSafe V5 devices use vendor-defined usage page **0xFF00, Magnetic Stripe Reader**.

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

- **Feature Reports**, which the host uses to send commands to the device. Feature reports can be further subdivided into **Get Feature** and **Set Feature** types. MagneSafe V5 devices only use one feature report.
- **Input Reports** are used by the device to send unsolicited notifications to the host when the device's state changes, or to send asynchronous responses to the host when a command completes. The device commonly uses input reports when reporting unpredictable cardholder interactions, or when a command takes more time for the device to process than is reasonable for the host to wait on a blocking call for the device to acknowledge completion.

For information about using feature reports to send commands to the device and receive responses from the device, see section **2.1.2 How to Send Commands On the USB Connection**. For information about receiving unsolicited data from the device via Input Reports, see section **2.1.3 How to Receive Data On the USB Connection (HID Only)**.

2.1.2 How to Send Commands On the USB Connection

Because many MagneSafe V5 devices support connection types beyond USB, this documentation abstracts host-device communication by referring to **Commands**, which are most often a pairing of a **Request** from the host and a corresponding **Response** from the device. This section explains how these terms apply when using the USB HID connection.

When the device is connected to the host via USB, regardless of whether it identifies and operates as a vendor-defined HID device or as a keyboard, the host sends a **Set Feature Report** to the device to send the requests for **Commands**, and sends a **Get Feature Report** to the device to retrieve a synchronous response when appropriate. All reports use Usage Page **0xFF00**, Usage ID **0x20**, and no Feature Report ID (Extended Commands Only) or, on devices that support Extended Commands, Feature report ID **0x01**.

The host should send both Feature Report types using the default Control pipe using a blocking call to the operating system's native USB libraries. The device NAKs the Status page of a **Set Feature Report** until it finishes the requested operation, and if it does not respond, the operating system will generally time out and report failure. This method ensures that as soon as the device has fulfilled the command request embedded in the **Set Feature Report**, the host software can immediately call a follow-up **Get Feature Report** to retrieve the command response, if one is required, and that the host software will not hang on a blocking call indefinitely.

The host should follow this general command sequence to send a request and receive a response:

- 1) Choose the command to invoke from section **8 Commands**. Every command has a corresponding **Command Number** listed in the header of its documentation section.
- 2) Construct a **Command Request Data** value using the Request table in the documentation for the command.
- 3) Determine the length of the **Command Request Data** value, referred to as the **Command Request Data Length**.
- 4) Examine the device's Report Descriptor to determine what payload length the device expects for a **Set Feature Report** and **Get Feature Report** (the operating system libraries may refer to this length as the "Report Length" or "Report Count").
- 5) Pad the **Command Request Data** value with 0x00 so the total length of the payload is consistent with the Set Feature Report's Report Length / Report Count.
- 6) Construct a Set Report Structure using **Table 8-1** in section **8.1 About Commands**.
- 7) Send a **Set Feature Report** containing the finalized padded Set Report Structure. The call to send the report may succeed, fail on a timeout, or fail for some other reason.
- 8) If the call succeeds, send a **Get Feature Report** to retrieve the device's response in the Get Report Structure shown in **Table 8-2** in section **8.1 About Commands**.
- 9) Parse the Get Report Structure, and truncate the **Command Response Data** field to the provided **Command Response Data Length**.
- 10) Examine the **Result Code**, which is a one-byte value the device sends to indicate success or the failure mode of the command. See section **8.2 About Result Codes** for more detail.
- 11) Parse the truncated **Command Response Data** field using the Response table in the documentation for the command.

In very rare cases, the host may simply send a **Get Feature Report** directly without a preceding **Set Feature Report**. The **Commands** documentation specifies these special cases if they exist.

(Extended Commands Only)

Commands that use two-byte Command Numbers are called **Extended Commands**. Generally these are commands that require a **Data Length** that is longer than the number of bytes available in a single report. The host must call these commands using **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**. Similarly, commands that send responses greater than the number of bytes available in a single report require the host to use **Command 0x4A - Get Extended Response (Extended Commands Only)** to retrieve Extended Responses. See the documentation for those two commands for details about how Extended Commands and Extended Responses work.

2.1.3 How to Receive Data On the USB Connection (HID Only)

When the device communicates with the host as a vendor-defined HID device, it sends unsolicited messages such as card data to the host via one or more **Input Reports**, which are asynchronous data packets (i.e., events) sent from the device to the host using the USB **Interrupt IN** pipe. Events occur when the device state changes or when an asynchronous command (such as a command that requires cardholder interaction) has reached a pre-defined event, such as completion. Per the USB HID standard, the host polls the device on a regular Polling Interval to see if it has input data available to send. If the device does not, it responds to the poll with a USB NAK.

Devices that do not support “Extended Notifications” (a specific way of sending asynchronous data to the host, see section **1.5 About Device Features**) implement a single input report for **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)**. Because these devices only implement one input report, the input report they send to the host does not include a report identifier, in accordance with the USB HID specification.

The host can locate a specific data element in the input report for **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)** by finding the corresponding Usage and interpreting its contents as binary data. For example, the host software can find **Track 1 Decode Status (HID | TLV | GATT | SLIP)** as follows:

- 1) Knowing from section **2.1.1** that the device uses usage page `0xFF00`, and knowing from the “Where to Find Value” column in the first table of section **6.2** that the desired data is found in the usage with Usage ID `0x0020`, call the platform’s USB SDK to retrieve the data from usage `0xFF000020` in the input report.
- 2) Interpret the single binary data byte from that Usage according to the second table in section **6.2**.

(Extended Notifications Only)

The remainder of this section provides important information about compatibility between host software designed for other MagneSafe V5 devices and this device.

Devices that support **Notification Messages Sent from Device to Host (Extended Notifications Only)** implement a HID input report specifically for sending notification message packets. Because the USB HID specification requires any device with more than one report of the same type to use HID report identifiers, such devices include a report identifier with every report:

- **Magnetic Stripe Card Data Sent from Device to Host** uses **Input Report ID 1** for card data. The card data report descriptor is typically included even if the device does not send card data, to allow for future flexibility.
- **Notification Messages Sent from Device to Host (Extended Notifications Only)** use **Input Report ID 2** for asynchronous notifications [see section 7].

Host software written for devices that support notification messages must specify these report identifiers when sending or retrieving reports to communicate with the device. Some pre-existing host software for Windows may expect to see report identifier zero, which the platform APIs may send when report IDs are not in use; this may need to change for compatibility with devices that use Extended Notifications.

The host can determine the size of notification message packet Input Reports by looking at the HID report descriptor. Notification message packet reports are generally 63 bytes long. If a notification message can’t fit into one packet, the device sends multiple packets, each containing the notification message packet format in **Table 7-2** and partial notification message data.

2 - Connection Types

The host can locate a specific data element in a notification input report by finding the corresponding Usage and interpreting its contents as binary data. For example, upon receiving an input report with **ID 2**, the host software can find the message payload as follows:

- 1) Knowing from section **2.1.1** that the device uses usage page `0xFF00`, and knowing from the “Where to Find Value” column in the first table of section **7.1** that the desired data is found in the usage with Usage ID `0x0020`, call the platform’s USB SDK to retrieve the data from usage `0xFF000020` in the input report.
- 2) Interpret the blob of data from that Usage according to the second table in section **7.1**.

2.2 How to Use Bluetooth LE Connections (Bluetooth LE Only)

This section provides information about developing software for a Bluetooth LE-capable host that needs to communicate with the device using Bluetooth Low Energy (Bluetooth LE). In this arrangement, the device acts as a Bluetooth LE server/peripheral, and the host acts as a client/central.

2.2.1 About GATT Characteristics

Table 2-1 - <DeviceName> GATT Service Characteristic

Characteristic Name	<DeviceName> GATT Service
Properties	Read
Data Size	N/A
UUID (LSB Order)	For eDynamo/tDynamo: 03:01:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Description/Usage	Used to identify the Service.

Table 2-2 - Command Data Characteristic

Characteristic Name	Command Data
Properties	Read/Write
Data Size	Variable (currently 60 bytes maximum but may increase).
UUID (LSB Order)	00:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Description/Usage	Contains the command data in USB HID feature report format without the fixed report size and padding (see Table 8-1 and Table 8-2 in section 2.1.2 How to Send Commands On the USB Connection for details). The data length field of the feature report is used to determine the length to be read or written. The length of the characteristic is 2 + the data length field value.

Table 2-3 - Card Data Characteristic

Characteristic Name	Card Data
Properties	Read, Notify
Data Size	Variable (512 max)
UUID (LSB Order)	01:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Description/Usage	<p>Contains the card data.</p> <p>If the host software configures the Card Data characteristic to enable notifications, the device sends card data to the host in multiple notification messages when a card is swiped. This is the fastest way for the device to transmit card data. The first byte of each notification message always contains the block identifier of the card data, starting with block 0 for the first message and incrementing in subsequent messages. The host software can use the block identifier field to detect whether blocks have been lost due to communication loss or out-of-range problems. The remaining bytes of each notification message contain card data. After the device has sent all card data, it sends one more notification message with block identifier 0xFF to indicate all the card data has been sent. This last notification message also contains a second byte indicating the total number of blocks of card data it transmitted.</p> <p>If the host configures the Card Data characteristic to disable notifications (default configuration), the device does not include block identifier fields in the blocks; the read simply fails if a communication error occurs. The host software must read the card data from the Card Data characteristic in blocks using long reads, and must use the Data Ready and Data Read Status characteristics.</p>

Table 2-4 - Data Ready Characteristic

Characteristic Name	Data Ready
Properties	Notify
Data Size	4
UUID (LSB Order)	02:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Description/Usage	<p>Contains the characteristic identifier (byte 0), characteristic block identifier (byte 1), and the block length (byte 2 and 3 LSB first) of the data that is ready to be read. The characteristic identifiers are defined as 0 = Command data, 1 = Card or notification data. The first block of card or notification data is block 0, the second block is block 1, and so on. The host software knows it has received all available data when the data block is less than 512 bytes long. If the last block of card data happens to be exactly 512 bytes long, the device sends an additional Data ready notification with a block length of zero.</p>

Table 2-5 - Data Read Status Characteristic

Characteristic Name	Data Read Status
Properties	Write
Data Size	3
UUID (LSB Order)	03:02:B6:0C:41:E3:43:F8:8F:89:82:AD:F8:E6:08:05
Description/Usage	Contains the characteristic identifier (byte 0), characteristic block identifier (byte 1), and the read status (byte 2) of the data that was ready to be read. The host software should write a 0 to this characteristic after reading a block of card data, to notify the device it is ready to read the next block of card data, at which point the device posts the next block of data to the Data ready characteristic. The device does not accept any more card swipes until the host writes to this characteristic. If the host fails to write to this characteristic within 10 seconds of being notified a card data block is ready, the device terminates the transaction and discards all card data.

2.2.2 How to Connect to a Device Using Bluetooth LE

The general steps for a host to communicate with the device via Bluetooth LE are as follows:

- 1) Scan for nearby Bluetooth LE peripherals advertising the desired GATT service UUID.
- 2) If multiple devices of the desired type are available, examine each device's name property. A specific device's default name is a constant, and by default is equal to the product name plus a hyphen plus the serial number on the device label.
- 3) Establish a Bluetooth LE connection with the device.
- 4) Pair with the device using passkey 000000. In many cases this step is operator-driven.
- 5) Make sure, if the host is expecting to receive data from any Bluetooth LE characteristics, those characteristics are configured to enable notifications (see section **2.2.1 About GATT Characteristics**). 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]`.
- 6) Send commands to the device (see section **2.2.3 How to Send Commands On the Bluetooth LE Connection**) and process incoming messages from the device (see section **2.2.4 How to Receive Data On the Bluetooth LE Connection**).

2.2.3 How to Send Commands On the Bluetooth LE Connection

To send a command request and to receive the command response, the host should do the following:

- 1) Make sure it is connected to the device (see section **2.2.2 How to Connect to a Device Using Bluetooth LE**).
- 2) Write the command request data to the **Command Data** characteristic (see section **2.2.1 About GATT Characteristics**).
- 3) Wait to receive a **Data Ready** notification with the characteristic identifier set to 0 (command data).
- 4) Read the command response data from the **Command Data** characteristic.
- 5) Interpret the data according to section **3.2 How to Use GATT Format (GATT Only)**.

For a full list of commands and details about how to use them, see section **8 Commands**.

2.2.4 How to Receive Data On the Bluetooth LE Connection

This section describes how the device sends unsolicited messages (messages that are not the direct response to a command) to the Bluetooth LE host. This includes **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)**, and **Notification Messages Sent from Device to Host (Extended Notifications Only)**.

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

In the normal operating mode for the device in GATT HID Vendor Defined mode, the device is always advertising when not connected. The Bluetooth LE host is responsible for optimizing the device's power consumption by only connecting when needed. If the Bluetooth LE host is not able to disconnect directly through its Bluetooth LE API, it can force the device to disconnect by using **Bluetooth LE Command 0x0B - Terminate Bluetooth LE Connection**.

To receive card data when the **Card Data** characteristic is configured to send notifications, the host software should do the following:

- 1) Make sure it is connected to the device (see section **2.2.2 How to Connect to a Device Using Bluetooth LE**).
- 2) Wait to receive a **Card Data** notification.
- 3) If the block identifier is 0xFF (no more card data), all card data has been received. Otherwise, continue to wait to receive more **Card Data** notifications.
- 4) Verify the number of card data blocks received equals the **number of card data blocks sent** field contained in the last notification message. A mismatch indicates a transmission error occurred.
- 5) Interpret the data according to section **3.2 How to Use GATT Format (GATT Only)**.

To receive card data when the **Card Data** characteristic is not configured to send notifications, the host should do the following:

- 1) Make sure it is connected to the device (see section **2.2.2 How to Connect to a Device Using Bluetooth LE**).
- 2) Wait to receive a **Data Ready** notification with the characteristic identifier set to 1 (card data).
- 3) If the **length** field of the **Data Ready** notification is greater than zero, read the block of card data from the **card data** characteristic.
- 4) Write the **data read status** characteristic with the characteristic identifier, block identifier, and read status of the card data block that is done being read.
- 5) If the length field of the data ready notification is less than 512, all data has been received. Otherwise, loop back to receive more **data ready** notifications with characteristic identifier set to 1.
- 6) Interpret the data according to section **3.2 How to Use GATT Format (GATT Only)**.

3 Data Formats

3.1 How to Use HID Format (HID Only)

When the device and host are communicating in vendor-defined HID mode, data comes from the device as described in section **2.1.3 How to Receive Data On the USB Connection (HID Only)**. The host software can retrieve the incoming data by examining the various usages in the report(s). For details about which usages to examine and how to interpret the data, see section **6 Magnetic Stripe Card Data Sent from Device to Host** for card data, and section **7 Notification Messages Sent from Device to Host (Extended Notifications Only)**.

3.2 How to Use GATT Format (GATT Only)

When operating as a vendor-defined GATT device, the device may send **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)** or **Notification Messages Sent from Device to Host (Extended Notifications Only)** in either normal or RLE format, depending on whether RLE would help compress the data or not. The host software should understand both formats.

Regardless of whether the device has GATT notifications enabled or disabled (see section 2.2.1 **About GATT Characteristics** and section 2.2.4 **How to Receive Data On the Bluetooth LE Connection**), the first byte of the card data block contains the GATT card data format field, which indicates what type of data it is sending and whether the data is RLE compressed as follows:

- 0x00 = **Card Data Normal**, which indicates the card data payload contains uncompressed card data in USB HID vendor defined report format (see section 6 **Magnetic Stripe Card Data Sent from Device to Host**).
- 0x01 = **Card Data RLE**, which indicates the card data payload contains run-length-encoded compressed card data in USB HID vendor-defined report format (see section 6 **Magnetic Stripe Card Data Sent from Device to Host** and the information below about RLE decoding).
- 0x02 = **Notification Uncompressed**, which indicates the payload contains an uncompressed notification message [see section 7 **Notification Messages Sent from Device to Host (Extended Notifications Only)**].
- 0x03 = **Notification RLE**, which indicates the payload contains a run-length-encoded compressed notification message [see section 7 **Notification Messages Sent from Device to Host (Extended Notifications Only)** and the information below about RLE decoding].

The device implements RLE as follows:

- 1) Any byte that is repeated more than once consecutively is run length encoded. Bytes that are not repeated stay as-is.
- 2) Repeated bytes are run-length encoded by repeating the byte twice, followed by the number of times the byte was repeated in the original data.
- 3) The maximum length of an encoded run is 255, so runs larger than 255 bytes are encoded as multiple runs of 255 bytes each until the last run.

For example, the data 0x44 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x05 0x66 0x00 0x00 is encoded as 0x44 0x55 0x55 0x09 0x66 0x00 0x00 0x02. A run of 260 0x00 bytes would be encoded as 0x00 0x00 0xFF 0x00 0x00 0x05.

The second and third byte of card data and notification data contain the uncompressed data payload field size in big endian order.

The fourth byte onward contains the data for the **Magnetic Stripe Card Data Sent from Device to Host** or **Notification Messages Sent from Device to Host (Extended Notifications Only)**.

The data size for command data and card data may increase with firmware updates, so the host software should be able to adapt to this. Adapting can be as simple as ignoring any extra data bytes that are not understood or expected.

If the **Card Data** characteristic (see section 2.2.1 **About GATT Characteristics**) is not configured to use notifications, the maximum notification message packet data length is the maximum characteristic size allowed by the Bluetooth LE specification (512), times the maximum number of block identifiers (256) = 131072 bytes minus headers (3 + 8) = 131061 bytes, which is large enough to fit a maximum sized notification message with a complete data length of 65535 bytes without splitting it into multiple packets.

If the **Card Data** characteristic (see section **2.2.1 About GATT Characteristics**) is configured to use notifications, the maximum notification message partial data length supported by the protocol is the maximum notification payload size (19), times the maximum number of block identifiers (255) = 4845 bytes - headers (3 + 8) = 4834 bytes.

4 Security Levels

Devices can be configured to operate at different Security Levels, which affects **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)**, the host software's ability to modify **Properties**, and the host software's ability to execute certain **Commands**. The Security Level can be increased by sending commands to the device, but can never be decreased. The sections below provide details about how each security level affects device behavior.

4.1 About Message Authentication Codes (MAC)

Commands in this manual that are tagged "MAC" are **privileged commands**. If the device is set to a Security Level higher than **Security Level 2**, the host software must calculate and append a four-byte Message Authentication Code ("MAC") to the Data field of the message, extending the length of the field by 4 bytes, to prove the sender is authorized to execute that command. If the device is set to **Security Level 2**, the device ignores the MAC field and the Device Serial Number field and the host can set them to all zeroes. If a MAC is required but not present or incorrect, the device returns 0x07.

In most cases, the host must calculate the MAC using the current DUKPT Key (which can be retrieved using **Command 0x09 - Get Current TDES DUKPT KSN** to get a reference to the key). In some cases, documented in the commands that are affected by it, the host must compute the MAC using the UIK installed in the device. In cases where only MagTek knows the UIK, MagTek must be involved to populate the MAC field.

The host must calculate the MAC over the whole command per *ISO 9797-1*, MAC Algorithm 3, Padding Method 1, using the **Message Authentication, request or both ways** variant as specified in *ANS X9.24-1:2009, Annex A*. Data supplied to the MAC algorithm should be provided in raw binary form, not converted to ASCII-hexadecimal.

Upon successfully completing a MACed command that used the DUKPT key, the device advances the DUKPT Key.

The serial number value included with MACs is always 16 bytes long. The 16th byte always contains 0x00. If the serial number is less than 15 bytes, it is left-justified and padded with binary zeroes.

4.2 Security Level 2

Security Level 2 is the least secure mode. In this mode, keys are loaded but the device does not require the host software to use them for most operations: Keys are used/needed to load new keys and to move to Security Level 3 or 4, but all other properties and commands are freely usable. The host can use **Command 0x15 - Get / Set Security Level (MAC)** to determine the device's current security level.

(MSR Only, HID Only)

In Security Level 2, if the device is using HID format [see section **3.1 How to Use HID Format (HID Only)**], the device sends data in the MagneSafe V5 format described in this manual or in USB HID SureSwipe format using the SureSwipe VID/PID, based on the setting in **Property 0x38 - HID SureSwipe Flag (SureSwipe Only, HID Only, MSR Only)**. For information about USB HID SureSwipe format, see *D99875191 Technical Reference Manual, USB HID SureSwipe & Swipe Reader*.

4.3 Security Level 3

At Security Level 3, many commands require security; most notably **Command 0x01 - Set Property (MAC)**. See section **4.1 About Message Authentication Codes (MAC)** for details. The host can use **Command 0x15 - Get / Set Security Level (MAC)** to determine the device's current security level.

Security Level 3 also enables encryption of data and inclusion of encrypted data where it may have been left out at a lower security level. For a list of specific data the device encrypts at this security level and how the host can decrypt it, see section **5 Encryption, Decryption, and Key Management**.

4.4 Security Level 4 (MSR Only)

When the device is at Security Level 4, the device requires the host to successfully complete an Authentication Sequence before it will transmit data from a magnetic stripe card swipe (see section **8.3.6 Command 0x10 - Activate Authenticated Mode**). Correctly executing the Authentication Sequence also causes the green LED to blink, alerting the operator that the device is being controlled by a host with knowledge of the keys—that is, an Authentic Host. The host can use **Command 0x15 - Get / Set Security Level (MAC)** to determine the device’s current security level.

4.5 Command Behaviors By Security Level

Table 4-1 shows the commands that are affected by the device’s security level. Commands that are not affected by the security level are not listed. The key is as follows:

- **Y** means the command can run at the specified security level.
- **N** means the command is prohibited at the specified security level.
- **C** means the customer may specify **Y** or **S** for that command when ordering.
- **S** means the command is secured [may require MACing, see section **4.1 About Message Authentication Codes (MAC)**].
- * indicates **Command 0x02 - Reset Device** has special behavior. If an Authentication sequence has failed, only a correctly MACed **Command 0x02 - Reset Device (MAC)** can be used to reset the device. This is to prevent a dictionary attack on the keys and to minimize a denial of service (DoS) attack.

Table 4-1 - Command Behaviors At Each Security Level

Command	Level 2	Level 3	Level 4 (MSR Only)
Any command that is not listed in this table works the same at all Security Levels.	Y	Y	Y
Command 0x01 - Set Property (MAC)	Y	S	S
Command 0x02 - Reset Device (MAC)	Y	*	*
Command 0x10 - Activate Authenticated Mode	N	Y	Y
Command 0x11 - Activation Challenge Response	N	Y	Y
Command 0x12 - Deactivate Authenticated Mode	N	Y	Y
Command 0x15 - Get / Set Security Level (MAC)	S	S	S
Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)	N	Y	Y
Extended Command 0x0302 - Cardholder Selection Result	N	Y	Y
Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)	N	Y	Y
Extended Command 0x0304 - Cancel Transaction (EMV Only)	N	Y	Y
Extended Command 0x0305 - Modify Terminal Configuration (MAC)	N	S	S

4 - Security Levels

Command	Level 2	Level 3	Level 4 (MSR Only)
Extended Command 0x0307 - Modify Application Configuration (MAC)	N	S	S
Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)	N	S	S
Extended Command 0x030C - Set Date and Time (MAC)	N	S	S
Extended Command 0x030E - Commit Configuration	N	Y	Y
Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)	N	S	S

5 Encryption, Decryption, and Key Management

5.1 About Encryption and Decryption

Some data exchanged between the device and the host is encrypted. This includes **Encrypted Track Data**, **Encrypted MagnePrint Data**, **Encrypted Session ID**, and parts of the **ARQC Messages (EMV Only)** and **Transaction Result Messages (EMV Only)**. To decrypt this data, the host must first determine what key to use, then decrypt the data.

5.2 How to Determine the Key

When the device and the host are using TDES DUKPT key management and the device is encrypting data (see **Security Levels**), the host software must do the following to generate a key (the “derived key”) to use for decryption:

- 1) **Determine the value of the Initial Key loaded into the device.** The lookup methods the host software uses depend on the overall solution architecture, and are outside the scope of this document. However, most solutions do this in one of two ways, both of which use the Initial Key Serial Number that arrives with the encrypted data (see **Command 0x09 - Get Current TDES DUKPT KSN** for details about interpreting the KSN):
 - a) Look up the value of the Base Derivation Key using the Initial KSN portion of the current KSN as an index value, then use TDES DUKPT algorithms to calculate the value of the Initial Key; or
 - b) Look up the value of the Initial Key directly, using the Initial KSN portion of the current KSN as an index value.
- 2) **Derive the current key.** Apply TDES DUKPT algorithms to the Initial Key value and the encryption counter portion of the KSN that arrives with the encrypted data.
- 3) **Determine which variant of the current key the device used to encrypt.** The variants are defined in *ANS X9.24-1:2009 Annex A*, which programmers of host software must be familiar with. Which variant the host should use depends on the type of data the host is decrypting or encrypting, and on device settings:
 - a) **Encrypted MagnePrint Data** is encrypted according to the setting in **Property 0x56 - MagnePrint Data Encryption Variant (MSR Only, Configurable MagnePrint Variants Only)**, if the device supports it. Otherwise, it is encrypted according to the setting in **Property 0x54 - Card Data Encryption Variant (MSR Only, Configurable MSR Variants Only)**, if the device supports it. Otherwise, it is encrypted using the **PIN Encryption variant**.
 - b) **Encrypted Track Data** and **Encrypted Session ID** is encrypted according to the setting in **Property 0x54 - Card Data Encryption Variant (MSR Only, Configurable MSR Variants Only)**, if the device supports it. Otherwise, it is encrypted using the **PIN Encryption variant**.
 - c) EMV data is encrypted according to the setting in **Property 0x67 - EMV Data Encryption Variant (EMV Only)**.
- 4) Use the variant algorithm with the current key to calculate that variant.
- 5) Decrypt the data according to the steps in section **5.3 How to Decrypt Data**.

5.3 How to Decrypt Data

For **Encrypted Track Data** and encrypted EMV data in **ARQC Messages (EMV Only)** and **Transaction Result Messages (EMV Only)**, the device begins by encrypting the first 8 bytes of clear text track data. The 8-byte result of this encryption is placed in an encrypted data buffer. The process continues using the DES CBC (Cipher Block Chaining) method with the encrypted 8 bytes XORed with the next 8 bytes of clear text. That result is placed in next 8 bytes of the encrypted data buffer, and the device continues until all clear text bytes have been encrypted. If the final block of clear text contains fewer than 8 bytes, the device pads the end of the block to make 8 bytes. After the final clear text block is XORed with the prior 8 bytes of encrypted data, the device encrypts it and places it in the encrypted data value. No Initial Vector is used in the process.

The host must decrypt the data in 8 byte blocks, ignoring any final unused bytes in the last block. When a value consists of more than one block, the host should use the CBC method to decrypt the data by following these steps:

- 1) Start decryption on the last block of 8 bytes (call it block N) using the key.
- 2) XOR the result of the decryption with the next-last block of 8 bytes (block N-1).
- 3) Repeat until reaching the first block.
- 4) Do not XOR the first block with anything.
- 5) Concatenate all blocks.
- 6) Determine the expected length of the decrypted data. In some cases this may be a standard field length, and in other cases the expected data length may accompany the encrypted data. When decrypting track data where no length is available, the host software can use the End Sentinel to find the actual end of the data (ignoring the padding at the end, which contains all zeroes).
- 7) Truncate the end of the decrypted data block to the expected data length, which discards the padding at the end.

6 Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)

The device sends card swipe data to the host even if it can not fully decode the data. How the host interprets incoming messages to find the data detailed in this section depends on the connection type (see section 2 **Connection Types**) and the data format (see section 3 **Data Formats**). Each subsection is tagged with the features, connection types, and data formats for which it is relevant. **Table 6-1** provides a convenient summary / index of all available values and their offsets.

Table 6-1 - List of Magnetic Stripe Data Sorted By GATT/SLIP Offset

Data	HID Usage	GATT/SLIP Offset
Track 1 Decode Status (HID TLV GATT SLIP)	0x20	0
Track 2 Decode Status (HID TLV GATT SLIP)	0x21	1
Track 3 Decode Status (HID TLV GATT SLIP, 3-Track Only)	0x22	2
Track 1 Encrypted Data Length (HID GATT SLIP)	0x28	3
Track 2 Encrypted Data Length (HID GATT SLIP)	0x29	4
Track 3 Encrypted Data Length (HID GATT SLIP, 3-Track Only)	0x2A	5
Card Encode Type (HID TLV GATT SLIP)	0x38	6
Track 1 Encrypted Data	0x30	7..118
Track 2 Encrypted Data	0x31	119..230
Track 3 Encrypted Data	0x32	231..342
MagnePrint Status	0x23	344..347
MagnePrint Data Length (HID GATT SLIP)	0x2B	348
Encrypted MagnePrint Data	0x33	349..476
Device Serial Number	0x40	477..492
Device Encryption Status	0x42	493..494
DUKPT Key Serial Number (KSN)	0x46	495..504
Track 1 Masked Data Length (HID GATT SLIP)	0x47	505
Track 2 Masked Data Length (HID GATT SLIP)	0x48	506
Track 3 Masked Data Length (HID GATT SLIP, 3-Track Only)	0x49	507
Track 1 Masked Data	0x4A	508..619
Track 2 Masked Data	0x4B	620..731
Track 3 Masked Data (3-Track Only)	0x4C	732..843
Encrypted Session ID	0x50	844..851
Track 1 Absolute Data Length (HID GATT SLIP)	0x51	852
Track 2 Absolute Data Length (HID GATT SLIP)	0x52	853

Data	HID Usage	GATT/SLIP Offset
Track 3 Absolute Data Length (HID GATT SLIP, 3-Track Only)	0x53	854
MagnePrint Absolute Data Length (HID TLV GATT SLIP)	0x54	855
Remaining MSR Transactions	0x55	856..858
MagneSafe Version Number (HID GATT SLIP)	0x56	859..866
SHA-1 Hashed Track 2 Data (HID TLV GATT SLIP, SHA-1 Only)	0x57	867...886
HID Report Version (HID GATT SLIP)	0x58	887
MagnePrint KSN (HID TLV GATT SLIP)	0x5A	920..929
Battery Level (HID GATT SLIP)	0x5B	930

6.1 About Track Data

After the host receives and decrypts **Encrypted Track Data** from the device, or receives clear text track data (based on device settings or state), or receives **Masked Track Data**, it may need to parse each track into individual values embedded in the tracks. The device can read multiple card formats, which vary even between different issuers and payment brands using the same underlying standards. Describing all possible formats is beyond the scope of this document, but this section describes how to parse data from tracks 1, 2, and 3 in a generic ISO/ABA compliant format as an example.

Table 6-2 shows an example of ISO/ABA track data the device sends to the host, using unmasked placeholder numbers to make it easier to see the relative positions of the values embedded in the track data. It is important to note that some cards do not include Track 3 data, and some devices do not read or transmit Track 3 data (see section **1.5 About Device Features**).

Table 6-2 – Example Generic ISO/ABA Track Data Format

Generic ISO/ABA Track Data Format	
Track 1 Data	%7555555555555555^CARDHOLDER NAME/^33338880004444000006?
Track 2 Data	;5555555555555555=33338880004444006?
Track 3 Data	;5555555555555555=333388800044440000006?

The example track data in **Table 6-2** can be interpreted as follows:

- The **%**, **?**, and **:** are Sentinels / delimiters, and are taken directly from the data on the card.
- The **7** at the beginning of Track 1 data is the card format code. For swiped credit / debit cards, this comes from the card and is generally **B**.
- The string of **5**s is the Account Number / License Number / PAN.
- The carets **^** are a standard ISO track 1 delimiter surrounding the Cardholder Name.
- **CARDHOLDER NAME/** is the Cardholder Name.
- The string of **3**s is the Expiration Date.
- The string of **8**s is the Service Code. For swiped credit / debit cards, this comes from the card.
- The remaining characters (**0**s, **4**s, and **6**) are Discretionary Data. For swiped debit / credit cards this data is of varying length and content and comes from the card, and must be interpreted according to the standards established by issuers, payment brands, and so on.

6.2 Track 1 Decode Status (HID | TLV | GATT | SLIP)

This one-byte value indicates the status of decoding Track 1. If bit 0 is OFF, no error occurred. If bit 0 is ON, the device found non-noise data that was not decodable, and the device reports the track data length is zero and does not provide valid track data to the host.

Format	Where to Find Value
HID	Usage 0x20
Streaming	N/A
TLV	Data Object 8262 Byte 1
GATT/SLIP	Offset 0

Bit Position	7	6	5	4	3	2	1	0
Value	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Error

6.3 Track 2 Decode Status (HID | TLV | GATT | SLIP)

This one-byte value indicates the status of decoding Track 2. If bit 0 is OFF, no error occurred. If bit 0 is ON, the device found non-noise data that was not decodable, and the device reports the track data length is zero and does not provide valid track data to the host.

Format	Where to Find Value
HID	Usage 0x21
Streaming	N/A
TLV	Data Object 8262 Byte 2
GATT/SLIP	Offset 1

Bit Position	7	6	5	4	3	2	1	0
Value	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Error

6.4 Track 3 Decode Status (HID | TLV | GATT | SLIP, 3-Track Only)

This one-byte value indicates the status of decoding Track 3. If bit 0 is OFF, no error occurred. If bit 0 is ON, the device found non-noise data that was not decodable, and the device reports the track data length is zero and does not provide valid track data to the host.

Format	Where to Find Value
HID	Usage 0x22
Streaming	N/A
TLV	Data Object 8262 Byte 3
GATT/SLIP	Offset 2

Bit Position	7	6	5	4	3	2	1	0
Value	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Error

6.5 Card Encode Type (HID | TLV | GATT | SLIP)

This one-byte value indicates the type of encoding the device found on a swiped magnetic stripe card. **Table 6-3** defines the possible values. For details about how the device determines the card's encode type, see **Appendix C Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only)**.

Format	Where to Find Value
HID	Usage 0x38
TLV	Data Object 8261
GATT/SLIP	Offset 6

Table 6-3 - Card Encode Types

Value	Encode Type	Description
0	ISO/ABA	ISO/ABA encode format. At least one track in ISO/ABA format, Track 3 not AAMVA format. See Appendix C Identifying ISO/ABA and AAMVA Cards for ISO/ABA description.
1	AAMVA	AAMVA encode format. Track 3 is AAMVA format, Tracks 1 and 2 are ISO/ABA if correctly decoded. See Appendix C Identifying ISO/ABA and AAMVA Cards for AAMVA description.
2	Reserved	Reserved.
3	Blank	The card is blank. All tracks decoded without error and without data.
4	Other	The card has a non-standard encode format. For example, ISO/ABA track 1 format on track 2.
5	Undetermined	The card encode type could not be determined because no tracks could be decoded. Combination of Error tracks and Blank Tracks, at least one Error track.

6.6 Device Encryption Status

This two-byte value contains the Device Encryption Status in big endian byte order. Byte 1 is the least significant byte; the LSB of byte 1 is status bit 0, and the LSB of byte 2 is status bit 15.

If the **Encryption Enabled** bit or **Initial DUKPT Key Injected** bit are not set, the device sends card data it would usually encrypt as clear text, and does not include a valid **DUKPT Key Serial Number**.

When the **DUKPT Keys Exhausted** bit is set, the device no longer reads cards, but continues to send **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)** to report status. The data it sends to the host in this case does not include valid **MagnePrint Status**, **Encrypted MagnePrint Data**, **Masked Track Data**, or **Encrypted Track Data**.

Format	Where to Find Value
HID	Usage 0x42
TLV	Data Object 8001
GATT/SLIP	Offset 493..494

The Device Encryption Status is defined as follows:

Bit	Meaning
0	DUKPT keys exhausted (1 = Exhausted, 0 = Keys available)
1	Initial DUKPT key injected, always set to 1
2	Encryption Enabled, always set to 1
3	Authentication Required
4	Timed out waiting for cardholder to swipe card
5	Reserved
6	Reserved
7	Reserved
8	No MSR Transactions Remaining [see Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only)]
9	(Secondary DUKPT Key Only) Initial Secondary DUKPT key injected
10	(Secondary DUKPT Key Only) DUKPT Key used to encrypt Encrypted Track Data , Encrypted Session ID . 0 = Primary, 1 = Secondary
11	(Configurable MSR Variants Only) DUKPT Key Variant used to encrypt Encrypted Track Data . 0 = PIN Encryption. 1 = Data Encryption, request or both ways
12	(Secondary DUKPT Key Only) DUKPT Key used to encrypt Encrypted MagnePrint Data , 0 = Primary DUKPT Key. 1 = Secondary DUKPT Key.

Bit	Meaning
13	(Configurable MagnePrint Variants Only) DUKPT Key Variant used to encrypt Encrypted MagnePrint Data . 0 = PIN Encryption, 1 = Data Encryption, request or both ways
14	Unused (always set to 0)
15	Unused (always set to 0)

6.7 Encrypted Track Data

If decodable track data exists for a given track, the device returns it in the corresponding **Track x Encrypted Data** value, described in the subsections below.

When the device is transmitting data in HID, GATT, or SLIP format, the **Encrypted Data** values are always 112 bytes long, which is the maximum number of bytes that can be encoded on a card. However, the length of actual valid data in each value may be less than 112 bytes, and is stored in the corresponding **Encrypted Data Length** value. The host software should ignore data located beyond the data length reported by the device.

The device decodes the data from each track on the card and converts it to ASCII, and includes all data starting with the start sentinel and ending with the end sentinel.

If the device is in a security level below **Security Level 3**, it sends the resulting track data in the **Track x Encrypted Data** values unencrypted. If the device is in **Security Level 3** or **Security Level 4**, it encrypts the data before sending. For information about how the device encrypts the data and how the host should decrypt it, see section **5 Encryption, Decryption, and Key Management**.

6.7.1 Track 1 Encrypted Data Length (HID | GATT | SLIP)

This one-byte value indicates the number of bytes in the **Track 1 Encrypted Data** value. The value is always a multiple of 8. If the value is 0, the device found no data on the track or encountered an error decoding the track.

After data is decrypted, there may be fewer bytes of decoded track data than indicated by this value. The number of bytes of decoded track data is indicated by the **Track 1 Absolute Data Length** value.

Format	Where to Find Value
HID	Usage 0x28
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 3

6.7.2 Track 2 Encrypted Data Length (HID | GATT | SLIP)

This one-byte value indicates the number of bytes in the **Track 2 Encrypted Data** value. The value is always a multiple of 8. If the value is 0, the device found no data on the track or encountered an error decoding the track.

After data is decrypted, there may be fewer bytes of decoded track data than indicated by this value. The number of bytes of decoded track data is indicated by the **Track 2 Absolute Data Length (HID | GATT | SLIP)** value.

Format	Where to Find Value
HID	Usage 0x29
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 4

6.7.3 Track 3 Encrypted Data Length (HID | GATT | SLIP, 3-Track Only)

This one-byte value indicates the number of bytes in the **Track 3 Encrypted Data** value. The value is always a multiple of 8. If the value is 0, the device found no data on the track or encountered an error decoding the track.

After data is decrypted, there may be fewer bytes of decoded track data than indicated by this value. The number of bytes of decoded track data is indicated by the **Track 3 Absolute Data Length** value.

Format	Where to Find Value
HID	Usage 0x2A
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 5

6.7.4 Track 1 Absolute Data Length (HID | GATT | SLIP)

This one-byte value indicates the number of usable bytes in the **Track 1 Encrypted Data** value after decryption. If the value is 0, the device found no data on the track or encountered an error decoding the track.

Format	Where to Find Value
HID	Usage 0x51
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 852

6.7.5 Track 2 Absolute Data Length (HID | GATT | SLIP)

This one-byte value indicates the number of usable bytes in the **Track 2 Encrypted Data** value after decryption. If the value is 0, the device found no data on the track or encountered an error decoding the track.

Format	Where to Find Value
HID	Usage 0x52
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 853

6.7.6 Track 3 Absolute Data Length (HID | GATT | SLIP, 3-Track Only)

This one-byte value indicates the number of usable bytes in the **Track 3 Encrypted Data** value after decryption. If the value is 0, the device found no data on the track or encountered an error decoding the track.

Format	Where to Find Value
HID	Usage 0x53
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 854

6.7.7 Track 1 Encrypted Data

For information about the contents of track data, see section **6.1 About Track Data**. For information about decryption, see section **5 Encryption, Decryption, and Key Management**.

Format	Where to Find Value
HID	Usage 0x30
TLV	Data Object 8215
GATT/SLIP	Offset 7..118

6.7.8 Track 2 Encrypted Data

For information about the contents of track data, see section **6.1 About Track Data**. For information about decryption, see section **5 Encryption, Decryption, and Key Management**.

Format	Where to Find Value
HID	Usage 0x31
TLV	Data Object 8216
GATT/SLIP	Offset 119..230

6.7.9 Track 3 Encrypted Data

On 2-track devices (see **Table 1-2 - Device Features**), this value is included in incoming data as a null value.

For information about the contents of track data, see section **6.1 About Track Data**. For information about decryption, see section **5 Encryption, Decryption, and Key Management**.

Format	Where to Find Value
HID	Usage 0x32
TLV	Data Object 8217
GATT/SLIP	Offset 231...342

6.8 MagnePrint Status

This four-byte value contains 32 bits of MagnePrint status information. The first byte, byte 1, is the least significant byte and its least significant bit is status bit 0. The final byte, byte 4, is the most significant byte and its most significant bit is status bit 31. For an example, see **Table 6-4** on page **52** which shows how to interpret MagnePrint Status bits for a value of A1050000.

If the device is set to a security level below **Security Level 3**, it uses the current value of **Property 0x15 - MagnePrint Flags** to determine the behavior of this value.

- Bit 0 = MagnePrint capable flag
- Bits 1 to 15 = Product revision & mode
- Bit 16 = Reserved
- Bit 17 = Reserved for noise measurement
- Bit 18 = Swipe too slow
- Bit 19 = Swipe too fast
- Bit 20 = Reserved
- Bit 21 = Actual card swipe direction (0 = Forward, 1 = Reverse)
- Bits 22-31 = Reserved

Format	Where to Find Value
HID	Usage 0x23
TLV	Data Object 8263
GATT/SLIP	Offset 344..347

Table 6-4 - MagnePrint Status Example for Value A1050000

Byte #	Nibble #	Bit #	Hex Value	Binary Value	Bit Meaning
Byte 1 = A1	1	7	A	1	Product Revision/Mode
		6		0	Product Revision/Mode
		5		1	Product Revision/Mode
		4		0	Product Revision/Mode
	2	3	1	0	Product Revision/Mode
		2		0	Product Revision/Mode
		1		0	Product Revision/Mode
		0		1	MagnePrint capable
Byte 2 = 05	3	15	0	0	Product Revision/Mode
		14		0	Product Revision/Mode
		13		0	Product Revision/Mode
		12		0	Product Revision/Mode
	4	11	5	0	Product Revision/Mode
		10		1	Product Revision/Mode
		9		0	Product Revision/Mode
		8		1	Product Revision/Mode
Byte 3 = 00	5	23	0	0	Reserved
		22		0	Reserved
		21		0	Direction
		20		0	Reserved
	6	19	0	0	Too Fast
		18		0	Too Slow
		17		0	Reserved for noise measurement
		16		0	Reserved
Byte 4 = 00	7	31	0	0	Reserved
		30		0	Reserved
		29		0	Reserved
		28		0	Reserved
	8	27	0	0	Reserved
		26		0	Reserved
		25		0	Reserved
		24		0	Reserved

6.9 MagnePrint Data Length (HID | GATT | SLIP)

This one-byte value indicates the number of bytes in the **Encrypted MagnePrint Data** value, which is always a multiple of 8 bytes in length. This value is zero if there is no MagnePrint data. After the Encrypted MagnePrint data is decrypted, there may be fewer bytes of MagnePrint data than indicated by this value. The number of bytes of decrypted MagnePrint data is indicated by **MagnePrint Absolute Data Length**.

If the device is set to a security level below **Security Level 3**, it uses the current value of **Property 0x15 - MagnePrint Flags** to determine the behavior of this value.

Format	Where to Find Value
HID	Usage 0x2B
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 348

6.10 MagnePrint Absolute Data Length (HID | TLV | GATT | SLIP)

This one-byte value indicates the number of usable bytes in **Encrypted MagnePrint Data** value after decryption.

If the device is set to a security level below **Security Level 3**, it uses the current value of **Property 0x15 - MagnePrint Flags** to determine the behavior of this value.

Format	Where to Find Value
HID	Usage 0x54
Streaming	N/A
TLV	Data Object 8263
GATT/SLIP	Offset 855

6.11 Encrypted MagnePrint Data

This value contains Encrypted MagnePrint data, which when decrypted generally yields a 54-byte value. The least significant bit of the first byte of data in the decrypted value corresponds to the first bit of MagnePrint data.

If the device is set to a security level below **Security Level 3**, it uses the current value of **Property 0x15 - MagnePrint Flags** to determine the behavior of this value.

To derive a decrypted MagnePrint value to authenticate a card, the host should do the following:

- 1) If the device transmitted a **MagnePrint Data Length**, truncate the data to that length to strip out protocol padding and yield a decryptable data block.
- 2) Decrypt the data block.
- 3) If the device transmitted a **MagnePrint Absolute Data Length (HID | TLV | GATT | SLIP)**, truncate the data to that length to yield the MagnePrint data.

Format	Where to Find Value
HID	Usage 0x33
TLV	Data Object 8218
GATT/SLIP	Offset 349..476

6.12 Device Serial Number

This 16-byte ASCII value contains the device serial number in a null-terminated string, so the maximum length of the device serial number, not including the null terminator, is 15 bytes. This device serial number can also be retrieved and set with **Property 0x03 - Device Serial Number**. This value is stored in non-volatile memory, so it persists when the device is power cycled.

Format	Where to Find Value
HID	Usage 0x40
TLV	Data Object 8102
GATT/SLIP	Offset 477..492

6.13 Masked Track Data

6.13.1 About Masking

If decodable track data exists for a given track, the device uses the **Track x Masked Track Data** value for that track to send a masked version of the data. The masked version includes one byte of data for each character decoded from the track, starting with the Start Sentinel and ending with the End Sentinel.

In masked track data, the device sends a specified mask character instead of the actual character read from the track. Which characters are masked depends on the **Card Encode Type (HID | TLV | GATT | SLIP)**: Only ISO/ABA (Financial Cards with *ISO/IEC 7813* Format code B) and AAMVA cards are selectively masked; all other card types are either sent entirely masked or entirely unmasked. More detail about masking is included in the sections below about each specific track.

There are separate masking settings for ISO/ABA format cards and AAMVA format cards (See **Property 0x07 - ISO Track Mask** and **Property 0x08 - AAMVA Track Mask** for more information). Each of these settings allows the host software to specify masking details for the Primary Account Number and Driver's License / ID Number (DL/ID#), the masking character to be used, and whether a correction should be applied to make the Mod 10 (Luhn algorithm) digit at the end of the PAN be correct.

Table 6-5 provides an example of data from tracks 1, 2, and 3 of a swiped ISO/ABA card after it has been decrypted or if the device has sent it in the clear. **Table 6-6** shows the same data as it might appear with a specific set of **Masked Track Data** rules applied.

Table 6-5 – Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted

Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted	
Track 1	%B6011000995500000^ TEST CARD ^15121015432112345678?
Track 2	;6011000995500000=15121015432112345678?
Track 3	;6011000995500000=1512101543211234567833333333333333333333333333333333?

Table 6-6 – Sample ISO/ABA Swiped Track Data, Masked

Sample ISO/ABA Swiped Track Data, Masked	
Track 1	%B6011000020000000^ TEST CARD ^151200000000000000000?
Track 2	;6011000020000000=151200000000000000000?
Track 3	;6011000020000000=00?

Data Formats with fixed Data field lengths (such as USB HID format, GATT format, and SLIP format, which are fixed at 112 bytes) include a **Masked Track Data Length** value for each track, which the host should use to truncate and ignore undefined data past the end of the track data. Formats where the host can easily determine where masked track data begins and ends (such as formats with delimiters or with data length built in to the format itself) do not include explicit masked track data lengths.

6.13.2 Track 1 Masked Data Length (HID | GATT | SLIP)

This one-byte value indicates how many bytes of decoded card data are in the **Track 1 Masked Data** value. This value is zero if there is no data on the track or if there was an error decoding the track.

Format	Where to Find Value
HID	Usage 0x47
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 505

6.13.3 Track 2 Masked Data Length (HID | GATT | SLIP)

This one-byte value indicates how many bytes of decoded card data are in the **Track 2 Masked Data** value. This value is zero if there was no data on the track or if there was an error decoding the track.

Format	Where to Find Value
HID	Usage 0x48
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 506

6.13.4 Track 3 Masked Data Length (HID | GATT | SLIP, 3-Track Only)

This one-byte value indicates how many bytes of decoded card data are in the **Track 3 Masked Data** value. This value is zero if there was no data on the track or if there was an error decoding the track.

Format	Where to Find Value
HID	Usage 0x49
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 507

6.13.5 Track 1 Masked Data

This value contains the masked track data for track 1. All characters are transmitted. For information about the contents of track data, see section **6.1 About Track Data**. For general information about masking, see section **6.13.1 About Masking** and **Appendix C Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only)**.

For an ISO/ABA card, the PAN is masked as follows:

- The number of initial characters and trailing characters specified by **Property 0x07 - ISO Track Mask** is sent unmasked. If Mod 10 correction is specified (see section **9.8 Property 0x07 - ISO Track Mask**), all but one of the intermediate characters of the PAN are set to zero; one of them is set such that the last digit of the PAN calculates an accurate Mod 10 check of the rest of the PAN as transmitted. If the Mod 10 correction is not specified, all of the intermediate characters of the PAN are set to the specified mask character.
- Cardholder Name and the Expiration Date are sent unmasked.

- The Service Code is always unmasked on newer devices.
- All Field Separators are sent unmasked.
- All other characters are set to the specified mask character.

For an AAMVA card, the specified mask character is substituted for all characters read from the card.

Format	Where to Find Value
HID	Usage 0x4A (112 bytes fixed, must be truncated)
TLV	Data Object 8221
GATT/SLIP	Offset 508..619

6.13.6 Track 2 Masked Data

This 112-byte value contains the masked track data for track 2. For general information about masking, see section **6.13.1 About Masking** and **Appendix C Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only)**.

For an ISO/ABA card, the PAN is masked as follows:

- The number of initial characters and trailing characters specified by **Property 0x07 - ISO Track Mask** is sent unmasked. If Mod 10 correction is specified (see **Property 0x07 - ISO Track Mask**), all but one of the intermediate characters of the PAN are set to zero; one of them is set such that the last digit of the PAN calculates an accurate Mod 10 check of the rest of the PAN as transmitted. If the Mod 10 correction is not specified, all of the intermediate characters of the PAN are set to the specified mask character.
- The Expiration Date is sent unmasked.
- The Service Code is always unmasked on newer devices.
- All Field Separators are sent unmasked.
- All other characters are set to the specified mask character.

For an AAMVA card, the DL/ID# is masked as follows:

- The specified number of initial characters are sent unmasked. The specified number of trailing characters are sent unmasked. If Mod 10 correction is specified (see **Property 0x08 - AAMVA Track Mask**), all but one of the intermediate characters of the DL/ID#PAN are set to zero; one of them is set such that last digit of the DL/ID# calculates an accurate Mod 10 check of the rest of the DL/ID# as transmitted. If the Mod 10 correction is not specified, all of the intermediate characters of the DL/ID# are set to the specified mask character.
- The Expiration Date and Birth Date are transmitted unmasked.
- All other characters are set to the specified mask character.

Format	Where to Find Value
HID	Usage 0x4B (112 bytes fixed, must be truncated)
TLV	Data Object 8222
GATT/SLIP	Offset 620-731

6.13.7 Track 3 Masked Data (3-Track Only)

This 112-byte value contains the Masked Track Data for track 3. On 2-track devices (see **Table 1-2 - Device Features**), this value is not included in the incoming data. For general information about masking, see section **6.13.1 About Masking** and **Appendix C Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only)**.

For an ISO/ABA card, the PAN is masked as follows:

- The number of initial characters and trailing characters specified by **Property 0x07 - ISO Track Mask** is sent unmasked. If Mod 10 correction is specified (see section **9.8 Property 0x07 - ISO Track Mask**), all but one of the intermediate characters of the PAN are set to zero; one of them is set such that last digit of the PAN calculates an accurate Mod 10 check of the rest of the PAN as transmitted. If the Mod 10 correction is not specified, all of the intermediate characters of the PAN are set to the specified mask character.
- All Field Separators are sent unmasked.
- All other characters are set to the specified mask character.

For an AAMVA card, the specified mask character is substituted for all characters read from the card.

Format	Where to Find Value
HID	Usage 0x4C (112 bytes fixed, must be truncated)
TLV	Data Object 8223
GATT/SLIP	Offset 732-843

6.14 Encrypted Session ID

This 8-byte value contains the encrypted version of the current Session ID. Its primary purpose is to prevent replays. After a card is read, this property is encrypted, along with the card data, and supplied as part of the transaction message. The clear text version is never transmitted. To avoid replay, the host software should set the Session ID property before a transaction, and verify that the Encrypted Session ID returned with card data decrypts to the original value it set.

Format	Where to Find Value
HID	Usage 0x50
TLV	Data Object 8309
GATT/SLIP	Offset 844..851

6.15 DUKPT Key Serial Number (KSN)

This 80-bit value contains the TDES DUKPT **Key Serial Number** (KSN) associated with encrypted values included in the same message. For details about how to interpret this value, see section **8.3.4 Command 0x09 - Get Current TDES DUKPT KSN**. If no keys are loaded, all bytes have the value 0x00.

Format	Where to Find Value
HID	Usage 0x46

Format	Where to Find Value
TLV	Data Object 8301
GATT/SLIP	Offset 495..504

6.16 Remaining MSR Transactions

This 3-byte value contains the number of MSR transactions remaining at the end of the current transaction. The value is also sometimes referred to as the transaction threshold. See **Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only)** for more information.

Format	Where to Find Value
HID	Usage 0x55
TLV	Data Object 810A
GATT/SLIP	Offset 856..858

6.17 MagneSafe Version Number (HID | GATT | SLIP)

This eight-byte value contains the MagneSafe Version Number with at least one terminating 0x00 byte to make string manipulation convenient. See **Property 0x04 - MagneSafe Version Number** for more information.

Format	Where to Find Value
HID	Usage 0x56
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 859..866

6.18 SHA-1 Hashed Track 2 Data (HID | TLV | GATT | SLIP, SHA-1 Only)

If the device supports SHA-1 (see **Table 1-2 - Device Features**), this 20-byte value contains a SHA-1 hash of either the PAN from track 2 or all the track 2 data, depending on the device's configuration stored in **Property 0x57 - SHA Hash Configuration (HID Only | TLV Only, Configurable SHA Only, MSR Only)**. If the device does not support SHA-1, this value is absent or contains padding.

Format	Where to Find Value
HID	Usage 0x57
Streaming	N/A
TLV	Data Object 8308
GATT/SLIP	Offset 867..886

6.19 HID Report Version (HID | GATT | SLIP)

This one-byte value identifies which variation of sets of values the device sends the host for **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)**.

If the data does not contain this value, the host should implicitly assume it is equal to 0x01. If the report does contain this value, it indicates the following:

HID Report Version	Changes
Empty	Original magnetic stripe card data contents
0x02	Added HID Report Version (HID GATT SLIP) Added SHA-256 Hashed Track 2 Data
0x03	Added Battery Level (HID GATT SLIP)

Format	Where to Find Value
HID	Usage 0x58
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 887

6.20 MagnePrint KSN (HID | TLV | GATT | SLIP)

This 80-bit value contains the TDES DUKPT Key Serial Number associated with encrypted MagnePrint values included in the same message. The rightmost 21 bits are the current value of the encryption counter. The leftmost 59 bits are a combination of the Key Set ID (KSID) that identifies the Base Derivation Key injected into the device during manufacture, and the device's serial number; how those two values are combined into the 59 bits is defined by a convention the customer decides when architecting the solution, with support from MagTek. If no keys are loaded, all bytes have the value 0x00.

Format	Where to Find Value
HID	Usage 0x5A
Streaming	N/A
TLV	Data Object 8305
GATT/SLIP	Offset 920..929

6.21 Battery Level (HID | GATT | SLIP)

This one-byte value contains the battery level of the device between 0% and 100%. 0x00 represents the lowest safe operating voltage; 0x64 means the battery is at full voltage. When the device is powered by USB, it always returns 100%. This field should be ignored for devices that do not contain a battery.

Format	Where to Find Value
HID	Usage 0x5B

Format	Where to Find Value
Streaming	N/A
TLV	N/A
GATT/SLIP	Offset 930

7 Notification Messages Sent from Device to Host (Extended Notifications Only)

7.1 About Notification Messages

This section provides detail about unsolicited generic notification messages the device sends to the host, excluding magnetic stripe card data documented separately in section **6 Magnetic Stripe Card Data Sent from Device to Host**. Each subsection is tagged with the features, connection types, and data formats for which it is relevant.

Notification messages may be split into multiple packets, each containing a portion of the complete notification message. This allows notification messages to exceed the maximum packet sizes of the connection type and data format. After the host receives a complete notification message, it will have a notification identifier, a complete data length, and a complete notification message data field.

How the host interprets incoming packets to find the data detailed in this section depends on the connection type (see section **2 Connection Types**) and the data format (see section **3 Data Formats**). All packets arrive at the host in the format-dependent structure shown in **Table 7-1**. Each incoming packet can be interpreted using **Table 7-2**. The **notification message** can be interpreted by first assembling all packets pertaining to the notification message, then looking up the corresponding **Notification Identifier** in the sections that follow.

Table 7-1 - How Notification Message Packets Arrive

Format	Where to Find Value
HID	Report identifier 2, Usage identifier 0x20
Streaming	N/A
TLV	N/A
GATT/SLIP	Similar to card data. For GATT, see section 2.2.4 How to Receive Data On the Bluetooth LE Connection and section 3.2 How to Use GATT Format (GATT Only) .

Table 7-2 - Structure of Packets That Form a Notification Message

Offset	Field Name	Description
0..1	Partial Data Length	The length of the Data field contained in the current message. This field is in big endian format. If this value is not equal to the Complete Data Length , the device is sending the notification using multiple packets.
2..3	Data Offset	The offset position in bytes within the entire assembled notification where the first byte of the current packet's Data field is located. This field is in big endian format. The first byte of the entire notification's Data is at offset zero.
4..5	Notification Identifier	The type of notification being sent. This field is in big endian format. The value corresponds to the notification identifier numbers in the headings of the subsections of section 7 Notification Messages Sent from Device to Host (Extended Notifications Only) . In many cases, two-byte notification identifiers are assigned such that the high byte indicates a group of related commands, and the low byte specifies a command within that group.

7 - Notification Messages Sent from Device to Host (Extended Notifications Only)

Offset	Field Name	Description
6..7	Complete Data Length	The total length of data for the entire notification message, summing all Partial Data Lengths for multiple packets. This field is in big endian format. If this value is not equal to the Partial Data Length of the current packet, the device is sending the data using multiple packets.
8..n	Data	May contain part or all of the notification data. The size of this field is contained in the Partial Data Length field.

7.2 Notification Group 0x03 - EMV L2 (EMV Only)

Notification Group 0x03 is reserved for EMV L2 notifications that support **Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)**. For more information about the general flow of EMV transactions, see section **8.4 Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)**.

7.2.1 Notification 0x0300 - Transaction Status / Progress Information

The device sends the host this notification to report progress during an EMV transaction the host has initiated using **Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)**. The granularity of notifications is designed to give specific information about transaction steps that involve interaction with either the cardholder or the host. More information about when the device sends this notification to the host can be found in the documentation for that command.

Some devices also send this notification outside the context of an EMV transaction to more generally notify the host that a card has been removed.

The behavior of this notification is partly driven by the settings in **Property 0x6D - EMV Contact Notification Configuration (Contact Only)**.

Notification Data

Offset	Field Name	Value
0	Event	Indicates the event that triggered this notification: 0x00 = No events since start of transaction 0x01 = Card Inserted (Contact Only) 0x02 = Payment Method Communication Error / Data Error 0x03 = Transaction Progress Change 0x04 = Waiting for Cardholder Response 0x05 = Timed Out 0x06 = End of Transaction 0x07 = Host Cancelled Transaction 0x08 = Card Removed (Contact Only)
1	Current Operation Time remaining	Indicates the remaining time available, in seconds, for the indicated operation to complete. The host specifies this timeout when calling Extended Command 0x0300 - Initiate EMV Transaction (EMV Only) .

7 - Notification Messages Sent from Device to Host (Extended Notifications Only)

Offset	Field Name	Value
2	Current Transaction Progress Indicator	<p>This one-byte field indicates the current processing stage for the transaction:</p> <p>0x00 = No Transaction In Progress 0x01 = Waiting for Cardholder to Present Payment 0x02 = Powering Up Card / Reading Magnetic Stripe 0x03 = Selecting the Application 0x04 = Waiting for Cardholder Language Selection (Contact Only) 0x05 = Waiting for Cardholder Application Selection 0x06 = Initiating Application 0x07 = Reading Application Data 0x08 = Offline Data Authentication 0x09 = Process Restrictions 0x0A = Cardholder Verification 0x0B = Terminal Risk Management 0x0C = Terminal Action Analysis 0x0D = Generating First Application Cryptogram 0x0E = Card Action Analysis 0x0F = Online Processing 0x10 = Waiting for Online Processing Response 0x11 = Transaction Complete 0x12 = Transaction Error 0x13 = Transaction Approved 0x14 = Transaction Declined 0x15 = Transaction Cancelled by MSR Swipe (MSR Only) 0x16 = EMV Error - Conditions Not Satisfied (Contact Only) 0x17 = EMV Error - Card Blocked (Contact Only) 0x18 = Contact Application Selection Failed (Contact Only) 0x19 = EMV Error - Card Not Accepted (Contact Only) 0x1A = Empty Candidate List 0x1B = Application Blocked 0x91 = Host Canceled EMV Transaction Before Card Was Presented</p>
3..4	Final Status	TBD

7.2.2 Notification 0x0301 - Display Message Request

The device sends this notification to request that the host display a message for the cardholder. The host should display the message.

Notification Data

Offset	Field Name	Value
0	Message	This is an array of bytes that should be displayed by the host on its display exactly as received. If the message is too long to fit on a single line it may be split to multiple lines if the host wishes. Messages are limited to 1024 bytes. If the message is zero length, this is a request for the host to clear the display.

7.2.3 Notification 0x0302 - Cardholder Selection Request (EMV Only)

This notification is used to inform the host that a cardholder selection is needed before the device can continue processing the current transaction. The host should prompt the cardholder to select an item from the menu, then send **Extended Command 0x0302 - Cardholder Selection Result** to inform the device that the transaction can proceed with the selected result.

Offset	Field Name	Value
0	Selection Type	This field specifies what kind of selection request this is: 0x00 = Application Selection 0x01 = Language Selection
1	Timeout	Specifies the maximum time, in seconds, allowed to complete the selection process. If this time is exceeded, the host should send Extended Command 0x0302 - Cardholder Selection Result with the Selection Status field set to 0x02 (Cardholder Selection Request aborted, timeout) after which the transaction is aborted and an appropriate Transaction Status is available. Value 0 (Cardholder Selection Request completed) is not allowed in this case.
2	Menu Items	This field is variable length and is a collection of null-terminated strings (maximum 17 strings). The maximum length of each string is 20 characters, not including a Line Feed (0x0A) character that may be in the string. The last string may not have the Line Feed character. The first string is a title and should not be considered for selection. It is expected that the host displays the menu items to the cardholder, then, after the cardholder makes a selection, call Extended Command 0x0302 - Cardholder Selection Result to return the number of the item the cardholder selected, which should be between 1 and the number of menu selection items being displayed. The first item, 0, is the title only.

7.2.4 Notification 0x0303 - ARQC Message

The device uses this notification to send ARQC data for the host to process. After the host processes the ARQC data, it should send **Extended Command 0x0303 - Online Processing Result / Acquirer Response** to inform the device it can proceed with the transaction.

Table 7-3 - Notification Data, ARQC Message

Offset	Field Name	Value
0	Message Length	Two byte binary, most significant byte first. This gives the total length of the ARQC message that follows, excluding padding and CBC-MAC.
2	ARQC Message	See Property 0x68 – EMV Message Format and Appendix D.1 ARQC Messages (EMV Only) . The host is expected to use this data to process a request.

7.2.5 Notification 0x0304 - Transaction Result Message

The device sends this notification to provide the host with final information from the transaction. It usually includes data and an indication of whether a signature is required.

Table 7-4 - Notification Data, Transaction Result Message

Offset	Field Name	Value
0	Signature Required	<p>This field indicates whether a cardholder signature is required to complete the transaction:</p> <p>0x00 = No signature required 0x01 = Signature required</p> <p>If a signature is required, the host should acquire the signature from the cardholder as part of the transaction data.</p>
1	Data Length	Two byte binary, most significant byte first. This gives the total length of the Data message that follows, excluding padding and CBC-MAC.
3	Data	See Appendix D.3 Transaction Result Messages and Property 0x68 – EMV Message Format . It is expected that the host will save this data as a record of the transaction.

8 Commands

This section describes the commands available on the device. Each command's section heading indicates the **Connection Types**, **Data Formats**, and device features (see section 1.5 **About Device Features**) that are relevant to it.

8.1 About Commands

Regardless of connection type and data format, all MagneSafe V5 devices use common structures to receive command request messages from the host and to send command response messages back. For information about connection-specific wrappers for these commands, see section 2 **Connection Types**.

Table 8-1 - Command Request Message (Host Sends to Device to Initiate a Command)

Offset	Field Name
0	Command Number
1	Command Request Data Length
2..n Maximum / fixed length depends on device and connection type.	Command Request Data

Command Number is a one byte value that contains the requested command number. Section 8 **Commands** lists all available commands.

Command Request Data Length is a one byte value that contains the length of the **Command Request Data** field.

Command Request Data contains command data as defined in the documentation for the selected command in section 8 **Commands**.

Table 8-2 - Command Response Message (Host Sends to Device to Retrieve Data or Responses)

Offset	Field Name
0	Result Code
1	Command Response Data Length
2..n	Command Response Data

Result Code is a one-byte value the device sends to indicate success or the failure mode of the command. Section 8.2 **About Result Codes** provides more detail.

Command Response Data Length is a one byte value that contains the length of the **Command Response Data** field.

Command Response Data contains response data as defined in the documentation for the selected command in section 8 **Commands**.

8.2 About Result Codes

There are two types of **Result Code** values the device can return in its response: **Generic** result codes (listed in **Table 8-3**), which have the same meaning for all commands, and **command-specific** result codes, which can have different meanings for different commands, and are listed with every command in this section. Generic result codes always have the most significant bit set to zero, and command-specific result codes always have the most significant bit set to one.

Table 8-3 - Generic Result Codes

Value (Hex)	Result Code	Description
0x00	Success	The command completed successfully.
0x01	Failure	The command failed.
0x02	Bad Parameter	The command failed due to a bad parameter or command syntax error.
0x03	Redundant	The command is redundant.
0x04	Bad Cryptography	A bad cryptography operation occurred.
0x05	Delayed	The request is refused because the device is delaying requests as a defense against brute-force hacking.
0x06	No Keys	No keys are loaded.
0x07	Invalid Operation	Depends on the context of the command.
0x08	Response not available	The response is not available.
0x09	Not enough power	The battery is too low to operate reliably.
0x0A	Extended response first packet (Extended Commands Only)	The device is returning the first (and possibly only) packet of an Extended Response.
0x0B	Extended command pending (Extended Commands Only)	An extended command is pending and the device is waiting for more data.
0x0C	Extended command notification (Extended Commands Only)	Deprecated
0x0D	Not implemented	The command is not implemented.
0x0E	Unarmed tamper, device not ready (Tamper Only)	The tamper device is not ready to be armed.
0x0F	Unarmed tamper, bad signature (Tamper Only)	The tamper is not armed because of a bad signature.

8.3 General Commands

8.3.1 Command 0x00 - Get Property

This command gets a property from the device. For details about properties, see section **9 Properties**.

Most properties have a firmware default value that may be changed during manufacturing or the order fulfillment process to support different customer needs.

Table 8-4 - Request Data for Command 0x00 - Get Property

Data Offset	Value
0	Property ID

Table 8-5 - Response Data for Command 0x00 - Get Property

Data Offset	Value
0..n	Property Value

Property ID is a one-byte value that identifies the property. A full list of properties can be found in section **9 Properties**.

Property Value consists of the multiple-byte value of the property. The number of bytes in this value depends on the type of property and the length of the property. **Table 8-6** describes the available property types.

Table 8-6 - Property Types

Property Type	Description
Byte	This is a one-byte value. The range of valid values depends on the property.
String	This is a null-terminated ASCII string. Its length can be zero to a maximum length that depends on the property. The length of the string does not include the terminating NULL character.

The result codes for the **Get Property** command can be any of the generic result codes listed in **Table 8-3** on page **71**.

8.3.2 Command 0x01 - Set Property (MAC)

This command sets a property in the device. For security purposes, this command is privileged. When the Security Level is set to higher than 2 (see section **4 Security Levels**), this command must be MACed to be accepted [see section **4.1 About Message Authentication Codes (MAC)**]. The command is logically paired with **Command 0x00 - Get Property**. For details about properties, see section **9 Properties**.

Some properties require the device to be reset using **Command 0x02 - Reset Device (MAC)** or power cycled to take effect. In those cases, the documentation for the property indicates what is required.

Table 8-7 - Request Data for Command 0x01 - Set Property (MAC)

Data Offset	Value
0	Property ID
1..n	Property Value

Response Data: None

The result codes for the **Set Property** command can be any of the generic result codes listed in **Table 8-3** on page **71**. If the **Set Property** command gets a result code of 0×07 , it means the required MAC was absent or incorrect.

Property ID is a one-byte value that identifies the property. A full list of properties can be found in section **9 Properties**.

Property Value consists of multiple bytes containing the value of the property. The number of bytes in this value depends on the property. **Table 8-4** describes the available property types.

Table 8-8 - Response Data for Command 0x01 - Set Property (MAC)

Property Type	Description
Byte	This is a one-byte value. The range of valid values depends on the property.
String	This is a multiple-byte ASCII string. Its length can be zero to a maximum length that depends on the property. The data length listed in the tables for each property does not include the terminating NULL character.

8.3.3 Command 0x02 - Reset Device (MAC)

This command is used to reset the device, and can be used to make property changes take effect without power cycling the device.

(USB Only)

When resetting a device that is using the USB connection, the device automatically does a USB Detach followed by an Attach. After the host sends this command to the device, it should close the USB port, wait a few seconds for the operating system to handle the device detach followed by the attach, then re-open the USB port before trying to communicate further with the device.

(PM3 | PM6 Only & Bluetooth LE Only)

When resetting a device that is using a Bluetooth LE connection, the device disconnects from Bluetooth LE. After the reset is complete, the device will be in airplane mode, and will not advertise over Bluetooth LE until either USB power is connected or the cardholder / operator presses and releases the button.

(MSR Only) If the device is in the midst of an Authentication Sequence initiated by **Command 0x10 - Activate Authenticated Mode (MSR Only)**, the device does not honor the Reset Device command until after the Authentication Sequence has successfully completed, or a cardholder swipes a card, or the device is power cycled. If the Authentication Sequence fails, the device initiates anti-hack mode and will require that the host MAC the Reset Device command (see section **4 Security Levels**). This prevents a dictionary attack on the keys and reduces the potential impact of denial of service attacks.

In rare instances, devices may optionally be configured at the manufacturer to require a MAC for every Reset Device command call, not just when anti-hack behavior is active.

Request Data Field: None

Response Data Field: None

Result codes:

0x00 = Success

0x07 = Incorrect MAC, or authentication sequence is pending

Example Request (Hex)

Cmd Num	Data Len	Data
02	00	

Example Response (Hex)

Result Code	Data Len	Data
00	00	

8.3.4 Command 0x09 - Get Current TDES DUKPT KSN

The host uses this command to get the current Triple Data Encryption Standard (TDES) DUKPT Key Serial Number (KSN) on demand.

This 80-bit value contains the TDES DUKPT **Key Serial Number** (KSN) associated with encrypted values included in the same message. The rightmost 21 bits are the current value of the encryption counter. The leftmost 59 bits are the device's **Initial KSN**, which is a combination of the **Key Set ID** that identifies the Base Derivation Key (BDK) injected into the device during manufacture, and the device's serial number (DSN); how those two values are combined into the 59 bit Initial KSN is defined by a convention the customer defines when architecting the solution, with support from MagTek. For example, one common scheme is to concatenate a 7 hex digit (28 bit) Key Set ID, a 7 hex digit (28 bit) Device Serial Number, and 3 padding zero bits. In these cases, the key can be referenced by an 8-digit MagTek part number ("key ID") consisting of the 7 hex digit Key Set ID plus a trailing "0."

Request Data: None

Table 8-9 - Response Data for Command 0x09 - Get Current TDES DUKPT KSN

Offset	Field Name	Description
0	Current Key Serial Number	80-bit TDES DUKPT KSN

Result codes:

0x00 = Success

0x02 = Bad Parameter - The Data field in the request is not the correct length. The request command contains no data, so the Data Length must be 0.

Example Request (Hex)

Cmd Num	Data Len	Data
09	00	None

Example Response (Hex)

Result Code	Data Len	Data
00	0A	FFFF 9876 5432 10E0 0001

8.3.5 Command 0x0A - Set Session ID (MSR Only)

This command is used to set the current Session ID, which the device transmits to the host in the **Encrypted Session ID**. The new Session ID stays in effect until one of the following occurs:

- The host sends the device another Set Session ID command.
- The device is powered off.
- The device is put into Suspend mode.

The Session ID is used by the host to uniquely identify the present transaction. Its primary purpose is to prevent replays. After the device reads a card, it encrypts the Session ID along with the card data, and supplies it as part of the **Magnetic Stripe Card Data Sent from Device to Host**. The device never transmits a clear text version of this data.

Table 8-10 - Request Data for Command 0x0A - Set Session ID (MSR Only)

Offset	Field Name	Description
0	New Session ID	This eight byte value may be any value the host software wishes.

Response Data: None

Result codes:

0x00 = Success

0x02 = Bad Parameter - The Data field in the request is not the correct length. The Session ID is an 8-byte value, so the Data Length must be 8.

Example Set Session ID Request (Hex)

Cmd Num	Data Len	Data
0A	08	54 45 53 54 54 45 53 54

Example Set Session ID Response (Hex)

Result Code	Data Len	Data
00	00	

8.3.6 Command 0x10 - Activate Authenticated Mode (MSR Only)

This command is used by the host software to activate Authenticated Mode, and is the only way to enter that mode. When the device is set to Security Level 4 (see section 4.4 Security Level 4), it does not gather and transmit card data after a swipe until Authenticated Mode has been established with the host, indicating both devices have established a direct two-way trust relationship. The general sequence of events for entering Authenticated Mode is as follows:

- 1) The cardholder or operator performs an action as a lead-in to swiping a card, such as signing in to a web page that interacts with the device.
- 2) The host software is aware of the cardholder action, and in response it sends the Activate Authenticated Mode command to the device. As part of this command, the host software specifies a PreAuthentication Time Limit parameter in units of seconds. The device uses this time limit in subsequent steps. The device interprets any value less than 120 seconds to mean 120 seconds.
- 3) The device responds to the host with the current Key Serial Number (KSN) and two challenges (Challenge 1 and Challenge 2), which it encrypts using a custom variant of the current DUKPT Key (Key XOR F0F0 F0F0 F0F0 F0F0 F0F0 F0F0 F0F0). Challenge 1 contains 6 bytes of random numbers followed by the last two bytes of the KSN. Challenge 2 contains 8 bytes of random numbers.
- 4) The device waits up to the PreAuthentication Time Limit. If the device times out waiting for the host to respond, the Authentication attempt fails and the device may activate anti-hacking behavior. See below for details.
- 5) The host software decrypts Challenge 1 and Challenge 2 and compares the last two bytes of the KSN with the last two bytes of the clear text KSN to authenticate the device.
- 6) The host software completes the Activate Authentication sequence using **Command 0x11 - Activation Challenge Response**, including the length of time the device should keep Authenticated Mode active without a swipe.
- 7) The device determines whether the Activation Challenge Reply is valid. If it is valid, the device activates Authenticated Mode and allows transmission of swiped card data to the host. The device may optionally indicate to the operator that the host and the device are mutually authenticated. See below for information about device behavior when the Activation Challenge Reply is not valid.
- 8) Authenticated mode stays active until the timeout previously specified by the host in **Command 0x11 - Activation Challenge Response**, or until the device sends valid swipe data to the host, at which point the device deactivates Authenticated Mode.

The first two Activate Authenticated Mode commands may proceed without any delay (one error is allowed with no anti-hacking consequences). If a second Activate Authenticated Mode in a row fails, the device activates anti-hacking behavior by enforcing an increasing delay between incoming Activate Authenticated Mode commands. The first delay is 10 seconds, increasing by 10 seconds up to a maximum delay of 10 minutes. The operator may deactivate anti-hacking mode at any time by swiping any encoded magnetic stripe card. When the device is in this anti-hacking mode, it requires the host to take additional steps to call **Command 0x02 - Reset Device**

To support use of Authenticated Mode, the host software can use **Command 0x14 - Get Device State (MSR Only)** at any time to determine the current state of the device.

Table 8-11 - Request Data for Command 0x10 - Activate Authenticated Mode (MSR Only)

Offset	Field Name	Description
0	PreAuthentication Time Limit (msb)	Most significant byte of the PreAuthentication Time Limit in seconds (120 seconds or greater)

Offset	Field Name	Description
1	PreAuthentication Time Limit (lsb)	Least significant byte of the PreAuthentication Time Limit in seconds (120 seconds or greater)

Table 8-12 – Response Data for Command 0x10 - Activate Authenticated Mode (MSR Only)

Offset	Field Name	Description
0	Current Key Serial Number	This eighty-bit value includes the Initial Key Serial Number in the leftmost 59 bits and the value of the encryption counter in the rightmost 21 bits.
10	Challenge 1	The host should use this eight-byte challenge later in Command 0x11 - Activation Challenge Response , and to authenticate the device.
18	Challenge 2	The host should use this eight-byte challenge later in Command 0x12 - Deactivate Authenticated Mode .

Result codes:

0x00 = Success

0x03 = Redundant - the device is already in this mode

0x05 = Delayed - the request is refused due to anti-hacking mode

0x07 = Sequence Error - the current Security Level is too low (see section 4 Security Levels)

0x80 = No MSR Transactions Remaining [see **Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only)**]

Example Request (Hex)

Cmd Num	Data Len	Data
10	00	

Example Response (Hex)

Result Code	Data Len	Data
00	1A	FFFF 0123 4567 8000 0003 9845 A48B 7ED3 C294 7987 5FD4 03FA 8543

8.3.7 Command 0x11 - Activation Challenge Response (MSR Only)

This command is used as the second part of an Activate Authentication sequence following **Command 0x10 - Activate Authenticated Mode**. In this command, the host software sends the first 6 bytes of Challenge 1 (received in response to **Command 0x10 - Activate Authenticated Mode**) plus two bytes of timeout information, and (optionally) an eight byte Session ID encrypted with the a custom variant of the current DUKPT Key (Key XOR 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C).

The time information contains the maximum number of seconds the device should remain in Authenticated Mode. Regardless of the value of this timer, a card swipe in the Authenticated Mode ends the Authenticated Mode. The maximum time allowed is 3600 seconds (one hour). For example, for a full hour, use 0x0E10; for 3 minutes, use 0x012C. A value of 0x00 forces the device to stay in Authenticated Mode until a card swipe or power down occurs (no timeout).

If the host includes Session ID information and the command is successful, it changes the Session ID in the device in the same way as calling **Command 0x0A - Set Session ID**.

If the device decrypts the Challenge Response correctly, Activate Authenticated Mode has succeeded. If the device can not decrypt the Challenge Response correctly, Activate Authenticated Mode fails and the TDES DUKPT Key Serial Number advances.

Table 8-13 - Request Data for Command 0x11 - Activation Challenge Response (MSR Only)

Offset	Field Name	Description
0	Response to Challenge 1	First 6 bytes of Challenge 1 plus a two-byte timeout (MSB first), encrypted by the specified variant of the current DUKPT Key.
8	Session ID	Optional eight byte Session ID encrypted by the specified variant of the current DUKPT Key.

Response Data: None

Result codes:

0x00 = Success

0x02 = Bad Parameters - the Data field in the request is not a correct length

0x04 = Bad Data - the encrypted reply data could not be verified

0x07 = Sequence - not expecting this command

Example Request (Hex)

Cmd Num	Data Len	Data
11	08	8579827521573495

Example Response (Hex)

Result Code	Data Len	Data
00	00	

8.3.8 Command 0x12 - Deactivate Authenticated Mode (MSR Only)

This command is used to exit Authenticated Mode initiated by **Command 0x10 - Activate Authenticated Mode**. It can be used to exit the mode with or without incrementing the DUKPT transaction counter (lower 21 bits of the Key Serial Number). The host software must send the first 7 bytes of Challenge 2 (from the response to **Command 0x10 - Activate Authenticated Mode**) and the Increment flag (0x00 indicates no increment, 0x01 indicates increment the KSN) encrypted with a custom variant of the current DUKPT Key (Key XOR 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C).

If the device decrypts Challenge 2 successfully, it exits Authenticated Mode, and depending on the Increment flag, may increment the KSN.

If the device cannot decrypt Challenge 2 successfully, it stays in Authenticated Mode until either the time specified in **Command 0x10 - Activate Authenticated Mode** elapses or the cardholder or operator swipes a card. This behavior is intended to discourage denial of service attacks. Exiting Authenticated Mode by timeout or card swipe always increments the KSN; exiting Authenticated Mode using **Command 0x12 - Deactivate Authenticated Mode** may increment the KSN.

Table 8-14 - Request Data for Command 0x12 - Deactivate Authenticated Mode (MSR Only)

Offset	Field Name	Description
0	Response to Challenge 2	Seven bytes of Challenge 2 plus one byte of Increment flag, encrypted by the specified variant of the current DUKPT Key

Response Data: None

Result codes:

0x00 = Success

0x02 = Bad Parameters - the Data field in the request is not the correct length

0x03 = Bad Data - the encrypted reply data could not be verified

0x07 = Sequence - not expecting this command

Example Request (Hex)

Cmd Num	Data Len	Data
12	08	8579827521573495

Example Response (Hex)

Result Code	Data Len	Data
00	00	

8.3.9 Command 0x14 - Get Device State (MSR Only)

When the device is set to **Security Level 4 (MSR Only)**, it requires mutual authentication with the host [see **Command 0x10 - Activate Authenticated Mode (MSR Only)**]. The host can use this command to determine the state of Authenticated Mode at a given point in time. For convenience, this manual refers to states with the notation *State:Antecedent* (e.g., **WaitActAuth:BadSwipe**), showing the current state and the state that led to it. Lists of possible states and their definitions are provided in the device response tables below.

In most cases, the host software can also track the state of Authenticated Mode by inference. As the host software interacts with the device, most state transitions are marked by the messages exchanged with the device. The exception is the transition from **WaitActRply:x** to **WaitActAuth:TOAuth**, which happens if the device times out waiting for the host to send **Command 0x11 - Activation Challenge Response (MSR Only)**, which the device does not report to the host. To cover this case, the host must be aware that a timeout could occur, in which case the device responds to **Command 0x11 - Activation Challenge Response (MSR Only)** with Result Code 0x07 (Sequence Error).

Example 1 – Power Up followed by Authentication and good swipe:

- 1) Device powers on. Host software should send this command to discover the current state of the device is **WaitActAuth:PU**.
- 2) Host sends a valid **Command 0x10 - Activate Authenticated Mode (MSR Only)**. Device responds with result code 0x00, inferring the transition to the **WaitActRply:PU** state.
- 3) Host sends a valid **Command 0x11 - Activation Challenge Response (MSR Only)**. Device responds with result code 0x00, inferring the transition to the **WaitSwipe:PU** state.
- 4) Cardholder swipes a card correctly. Device sends card data to the host, inferring the transition to the **WaitActAuth:GoodSwipe** state.

Example 2 – Device times out waiting for swipe:

- 1) Device waiting after a good swipe. Host software may send this command to discover the current state of the device is **WaitActAuth:GoodSwipe**.
- 2) Host sends valid **Command 0x10 - Activate Authenticated Mode (MSR Only)**. Device responds with result code 0x00, inferring the transition to the **WaitActRply:GoodSwipe** state.
- 3) Host sends a valid **Command 0x11 - Activation Challenge Response (MSR Only)**. Device responds with result code 0x00, inferring the transition to the **WaitSwipe:GoodSwipe** state.
- 4) Authenticated mode times out before a swipe occurs. Device sends mostly empty card data to the host to report the timeout in Device Encryption Status. The host infers the transition to the **WaitActAuth:TOSwipe** state.

Example 3 – Host sends invalid Command 0x11 - Activation Challenge Response (MSR Only):

- 1) Device waiting after a good swipe. Host software may send this command to discover the current state of the device is **WaitActAuth:GoodSwipe**.
- 2) Host sends valid **Command 0x10 - Activate Authenticated Mode (MSR Only)**. Device responds with result code 0x00, inferring the transition to the **WaitActRply:GoodSwipe** state.
- 3) Host sends invalid **Command 0x11 - Activation Challenge Response (MSR Only)**. Device responds with result code 0x02 or 0x04, inferring the transition to the **WaitActAuth:FailAuth** state.

Example 4 – Host waits too long sending Command 0x11 - Activation Challenge Response (MSR Only):

- 1) Device waiting after a good swipe. Host software may send this command to discover the current state of the device is **WaitActAuth:GoodSwipe**.

8 - Commands

- 2) Host sends valid **Command 0x10 - Activate Authenticated Mode (MSR Only)**. Device responds with result code 0x00, inferring the transition to the WaitActRply:GoodSwipe state.
- 3) Device times out waiting for host to send **Command 0x11 - Activation Challenge Response (MSR Only)** (State => WaitActAuth:TOAuth). Host doesn't know because the device does not send any message.
- 4) Host eventually sends **Command 0x11 - Activation Challenge Response (MSR Only)** (State remains WaitActAuth:TOAuth). Device responds with result code 0x07, inferring the previous transition to WaitActAuth:TOAuth state.

Request Data: None

Table 8-15 - First Byte, Response Data for Command 0x14 - Get Device State (MSR Only)

Current Device State		
Value	Name	Meaning
0x00	WaitActAuth	Waiting for Activate Authenticated Mode. The device requires the host to authenticate using Command 0x10 - Activate Authenticated Mode before it accepts swipes.
0x01	WaitActRply	Waiting for Activation Challenge Reply. The host has started to authenticate, and the device is waiting for Command 0x11 - Activation Challenge Response .
0x02	WaitSwipe	Waiting for swipe. The device is waiting for the cardholder or operator to swipe a card.
0x03	WaitDelay	Waiting for Anti-Hacking Timer. Two or more previous attempts to Authenticate have failed; the device is waiting for the Anti-Hacking timer to expire before it accepts Command 0x10 - Activate Authenticated Mode .

Table 8-16 - Second Byte, Response Data for Command 0x14 - Get Device State (MSR Only)

Current State Antecedent		
Value	Name	Meaning
0x00	PU	Just Powered Up. The device has had no swipes and has not been Authenticated since it was powered up.
0x01	GoodAuth	Authentication Activation Successful. The host has sent the device a valid Command 0x11 - Activation Challenge Response .
0x02	GoodSwipe	Good Swipe. The cardholder swiped a valid card correctly.
0x03	BadSwipe	Bad Swipe. The cardholder swiped a card incorrectly or the card is not valid.
0x04	FailAuth	Authentication Activation Failed. The most recent Command 0x11 - Activation Challenge Response failed.
0x05	FailDeact	Authentication Deactivation Failed. A recent Command 0x12 - Deactivate Authenticated Mode failed.
0x06	TOAuth	Authentication Activation Timed Out. The host failed to send Command 0x11 - Activation Challenge Response in the time period specified by Command 0x10 - Activate Authenticated Mode .

8 - Commands

Current State Antecedent		
0x07	TOSwipe	Swipe Timed Out. The cardholder failed to swipe a card in the time period specified in Command 0x11 - Activation Challenge Response .
0x08	Reserved	Reserved

Result codes:

0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
14	00	

Example Response (Hex)

Result Code	Data Len	Data
00	02	00 00

8.3.10 Command 0x15 - Get / Set Security Level (MAC)

This command is used to set or get the device's current Security Level (see section 4 Security Levels). The host can use this to raise the Security Level, but can not lower it.

When using this command to set the device's security level, the host should include the specified data in the request, and the device will not return an explicit response. When using this command to get the device's current security level, the host should include no data, and the device will return a response.

Table 8-17 - Request Data for Command 0x15 - Get / Set Security Level (MAC)

Offset	Field Name	Description
0	Security Level	Optional: if present must be either 0x03 or 0x04. If absent, this is a query for the current Security Level.
1	MAC	Four byte MAC to secure the command [see section 4.1 About Message Authentication Codes (MAC)]. If the host does not include a value for Security Level, it should not include the MAC value.

Table 8-18 - Response Data for Command 0x15 - Get / Set Security Level (MAC)

Offset	Field Name	Description
0	Security Level	Only present if there was no Data in the request. This value gives the current Security Level.

Result codes:

0x00 = Success

0x02 = Bad Parameters. The Data field in the request is not a correct length OR the specified Security Level is invalid; OR the current Security Level is 4.

0x07 = Incorrect MAC; command not authorized

Example Set Security Level Request (Hex)

Cmd Num	Data Len	Data
15	05	03 xx xx xx xx, where xx xx xx xx is a valid MAC

Example Set Security Level Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Data
15	00	

Example Get Security Level Response (Hex)

Result Code	Data Len	Data
00	01	03

8.3.11 Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only)

This command is used to get the maximum number of remaining card swipe transactions or correctly completed Authentication sequences (**Command 0x10 - Activate Authenticated Mode** followed by a correct **Command 0x11 - Activation Challenge Response**) that the device can process. The value it returns is also sometimes referred to as the transaction threshold.

The value has three possible states:

- **Disabled** - value 0xFFFFFFFF - In this state there is no limit to the number of transactions that can be performed.
- **Expired** - value 0x000000 - This state indicates MSR transactions and Authentication commands are prohibited.
- **Active** - value 1 to 1,000,000 (0x000001 to 0x0F4240) - In this state, each transaction or successful Authentication sequence causes the value to be decremented and allows transactions to be processed. If an Authentication sequence decrements this value to 0, the device permits one final encrypted card swipe.

Some devices are configured to only allow the manufacturer to call this command.

Request Data: None

Table 8-19 - Response Data for Command 0x1C - Get Remaining MSR Transactions Counter (MSR Only)

Offset	Field Name	Description
0	Device Serial Number	16 bytes of device serial number. If the serial number is shorter than 15 bytes, this value is left-justified and padded with binary zeroes. At least one byte (usually the last one) must contain binary zero.
16	Remaining MSR Transactions	This three byte value contains the current number of remaining MSR transactions.

Result codes:

0x00 = Success

0x02 = Invalid length

Example Request (Hex)

Cmd Num	Data Len	Data
1C	00	

Example Response (Hex)

Result Code	Data Len	Data
00	13	544553542053455455502030303031000007F1 (2033 MSR Transactions Remaining)

8.3.12 Command 0x45 - Get Battery Percentage (PM3 Only | PM4 Only | PM5 Only | PM6 Only | PM7 Only)

This command is used to get the percentage of useful battery charge remaining, in a range between 0x00 (0%) and 0x64 (100%).

Request Data: None

Table 8-20 - Response Data for Command 0x45 - Get Battery Percentage (PM3 Only | PM4 Only | PM5 Only | PM6 Only | PM7 Only)

Offset	Field Name	Description
0	Battery Percentage	

Result codes:

0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
45	00	

Example Response (Hex)

Result Code	Data Len	Data
00	01	62 (Battery at 98%, almost full charge)

8.3.13 Command 0x46 - Send Command to Bluetooth LE Controller (Bluetooth LE Only)

This command sends commands to the device's Bluetooth LE controller, which has its own command set used to control Bluetooth LE-specific aspects of the device. The valid command identifiers and data are defined in the following subsections.

Table 8-21 - Request Data for Command 0x46 - Send Command to Bluetooth LE Controller (Bluetooth LE Only)

Offset	Field Name	Description
0	Data type	Always set to 1 (control data).
1	Message type	Always set to 0 (request).
2	Bluetooth LE command identifier	The identifier of the Bluetooth LE command.
3..n	Bluetooth LE command request data	The data associated with the Bluetooth LE command request.

Table 8-22 - Response Data for Command 0x46 - Send Command to Bluetooth LE Controller (Bluetooth LE Only)

Offset	Field Name	Description
0	Data type	Always 1 (control data).
1	Message type	Always 1 (response).
2	Bluetooth LE result code	A code that indicates the result of the Bluetooth LE command. Valid values for this code are 0 for success, 1 for failure and 2 for bad parameter.
3..n	Bluetooth LE command response data	The data associated with the Bluetooth LE command response.

Result codes:

0x00 = Success

0x01 = Fail (timed out waiting for a response)

Example Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 02 01 02 03 (Send Echo command)

Example Response (Hex)

Result Code	Data Len	Data
00	06	01 01 00 01 02 03

8.3.13.1 Bluetooth LE Command 0x00 - Get Property

This command gets Bluetooth LE controller properties. The properties are listed in **Appendix A Bluetooth LE Controller Properties**.

Table 8-23 - Request Data

Byte offset	Field name	Description
0	Property identifier	The identifier of the property.

Table 8-24 - Response Data

Byte offset	Field name	Description
0..n	Property value	The value of the property.

8.3.13.2 Bluetooth LE Command 0x01 - Set Property

This command sets Bluetooth LE controller properties. The properties are listed in **Appendix A Bluetooth LE Controller Properties**.

Table 8-25 - Request Data

Byte offset	Field name	Description
0	Property identifier	The identifier of the property.
1..n	Property value	The value of the property.

Response Data: None

8.3.13.3 Bluetooth LE Command 0x02 - Echo

This is a testing command that echoes the data received in the request by transmitting it back to the host as a response.

Table 8-26 - Request Data

Byte offset	Field name	Description
0 - N	Echo data	Data to echo

Table 8-27 - Response Data

Byte offset	Field name	Description
0 - N	Echo data	Data echoed

Example Request (Hex)

Command Identifier	Request Data Length	Request Data
46	06	01 00 02 01 02 03 (echo 01 02 03)

Example Response (Hex)

Result Code	Response Data Length	Response Data
00	06	01 01 00 01 02 03

8.3.13.4 Bluetooth LE Command 0x06 - Erase All Non-Volatile Memory

This command erases the Bluetooth LE module's non-volatile memory, which returns it to its un-configured factory default state. This includes erasing all bonds (see **Bluetooth LE Command 0x07 - Erase All Bonds**). The command requires the host software to include a pair of Secure Code values in the request to make sure the host software does not accidentally invoke this command.

After calling this command, either the host must send **Command 0x02 - Reset Device** or a cardholder / operator must power it off for at least 30 seconds, then power it on, before the changes will take effect. Because this property affects Bluetooth LE communication, it is best to send it using the USB connection.

Table 8-28 - Request Data

Byte offset	Field name	Description
0	Secure code 1	Set to 0x55
1	Secure code 2	Set to 0xAA

Response Data: None

Example Request (Hex)

Command identifier	Request data length	Request data
46	05	01 00 06 55 AA

Example Response (Hex)

Result code	Response data length	Response data
00	03	01 01 00

8.3.13.5 Bluetooth LE Command 0x07 - Erase All Bonds

This command clears all pairing information about known Bluetooth LE hosts from the device. After issuing this command, unpair the device from all paired Bluetooth LE hosts prior to trying to re-pair the device. If any previously paired Bluetooth LE hosts are still in range of the device after issuing this command, they may try to re-connect to the device, which would cause the device to stop advertising and render it unable to re-pair. After clearing the device from all Bluetooth LE hosts, re-pair with the desired Bluetooth LE host(s).

The command requires the host software to include a pair of Secure Code values in the request to make sure the host software does not accidentally invoke this command.

After calling this command, either the host must send **Command 0x02 - Reset Device** or a cardholder / operator must power it off for at least 30 seconds, then power it on, before the changes will take effect. Because this property affects Bluetooth LE communication, it is best to send it using the USB connection.

Table 8-29 - Request Data

Byte offset	Field name	Description
0	Secure code 1	Set to 0x55
1	Secure code 2	Set to 0xAA

Response Data: None

Example Request (Hex)

Command identifier	Request data length	Request data
46	05	01 00 07 55 AA

Example Response (Hex)

Result code	Response data length	Response data
00	03	01 01 00

8.3.13.6 Bluetooth LE Command 0x0B - Terminate Bluetooth LE Connection

This command signals the device to wait 1 second then terminate the specified Bluetooth LE connection. The delay allows time for the host software to receive a response from the device if the command is issued over Bluetooth LE. To conserve battery power, the Bluetooth LE host should terminate the Bluetooth LE connection when it does not need to communicate to the device. Instead of using this command, the Bluetooth LE host may also directly terminate the Bluetooth LE connection if it is capable.

Request Data: None

Response Data: None

Example Request (Hex)

Command identifier	Request data length	Request data
46	03	01 00 0B

Example Response (Hex)

Result code	Response data length	Response data
00	03	01 01 00

8.3.13.7 Bluetooth LE Command 0x0D - Get Bond Count (Pairing Modes Only)

This command can be used to retrieve the number of hosts currently bonded to the device. The device can bond with up to the maximum number of bonds specified in **Bluetooth LE Property 0x16 - Maximum Bond Count (Pairing Modes Only)**.

Request Data: None

Response Data: One byte that contains the number of hosts currently bonded to the device.

Example Request (Hex)

Command identifier	Request data length	Request data
46	03	01 00 0D

Example Response (Hex)

Result code	Response data length	Response data
00	04	01 01 00 03 (03 = 3 bonds)

8.3.14 Command 0x48 - Notification Output Connection Override (Bluetooth LE Only | iAP Only, USB Only)

The host uses this command to immediately and temporarily override the current setting in **Property 0x5F - Notification Output Connection (Bluetooth LE Only | iAP Only, USB Only)** until the device is power cycled or reset, changing the connection the device uses to send **Magnetic Stripe Card Data Sent from Device to Host** [see section 2 Connection Types] and **Notification Messages Sent from Device to Host (Extended Notifications Only)** to the host.

If the host does not specify a connection type in the request, the response's Connection value returns the current connection type, otherwise the response contains no additional data.

Table 8-30 - Request Data for Command 0x48 - Notification Output Connection Override (Bluetooth LE Only | iAP Only, USB Only)

Offset	Field Name	Description
0	Connection	0x00 = USB 0x01 = Bluetooth LE (Bluetooth LE Only)

Table 8-31 - Response Data for Command 0x48 - Notification Output Connection Override (Bluetooth LE Only | iAP Only, USB Only)

Offset	Field Name	Description
0	Connection	0x00 = USB 0x01 = Bluetooth LE (Bluetooth LE Only)

Result codes:
0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
48	01	01

Example Response (Hex)

Result Code	Data Len	Data
00	00	

8.3.15 Command 0x49 - Send Extended Command Packet (Extended Commands Only)

The host uses this command to send **extended commands** to the device as one or more data packets. This **extended commands protocol** doubles the command number namespace to two bytes, doubles the result code namespace to two bytes, and supports commands and responses which require larger data payloads than those available for standard commands (shown in **Table 8-1** and **Table 8-2** in section **8.1 About Commands**).

If the required command data is 52 bytes or shorter, the host can send the entire command using a single extended command packet. If the command data is longer than 52 bytes, the host should split the data into multiple packets of 52 or fewer bytes, and send multiple extended command packets. Assuming 52-byte packets, the first packet the host sends should specify Extended Data Offset = 0, the next packet should specify Extended Data Offset = 52, and so on, until the host has sent all the command data. The device's response to each packet contains either an extended command result code or a standard result code for the command that was sent:

- **Result Code 0x0B - Extended Protocol Request Pending** indicates the device is buffering the incoming data and expects the host to send subsequent packets.
- **Result Code 0x0A - Extended Command Response** indicates the device has received the complete data set and has executed the command. If the device has 52 bytes or fewer to return to the host, that concludes the round trip of the command. If the response data is greater than 52 bytes, the host must retrieve additional data by continuing to call **Command 0x4A - Get Extended Response** until it has retrieved all response data.
- **Standard Result Code.** When using this command to invoke a standard command (as opposed to an extended command), see the Result Codes in the documentation for the command the host is invoking.

To simplify the development of custom host software, developers who are working exclusively with devices that support extended commands may choose to send all commands, including the single-byte commands described in this manual, using the extended commands protocol.

Table 8-32 - Request Data for Command 0x49 - Send Extended Command Packet (Extended Commands Only)

Offset	Field Name	Description
0..1	Extended Data Offset	This field is in big endian format. It indicates the byte offset position of this packet's Extended Data field, relative to the complete extended data field being sent as multiple packets. The Extended Data Offset of Packet 0 is 0.
2..3	Extended Command Number	This field is in big endian format and contains the number of the command to execute. For one-byte command numbers, the high byte should be set to zero.
4..5	Complete Extended Data Length	This field is in big endian format and gives the total length of the Extended Data field the host is sending as multiple packets.
6..n	Extended Data	This field contains either part or all of the extended data request the host is sending to the device. The size of this Extended Data field can be determined by subtracting the Extended Data field's offset within the request (6) from the request's total data length (N). In most cases the request's complete data payload can have a maximum value of 58 (for example see section 2.1.2 How to Send Commands On the USB Connection), so this field can have a maximum length of $58 - 6 = 52$ bytes.

Table 8-33 - Response Data for Command 0x49 - Send Extended Command Packet (Extended Commands Only)

Offset	Field Name	Description
0..1	Extended Data Offset	This field is in big endian format. It indicates the byte offset position of this packet's Extended Data field, relative to the complete extended data field being sent as multiple packets. The first byte is offset zero.
1..2	Extended Result Code	This field is in big endian format. For one-byte result codes, the high byte is set to zero.
4..5	Complete Extended Data Length	This field is in big endian format and gives the total length of the extended data field the host is sending as multiple packets.
6..n	Extended Data	This field contains either part or all of the complete Extended Data response the device is sending to the host. The size of this Extended Data field can be determined by subtracting the Extended Data field's offset within the response (6) from the response's total data length (N). In most cases the response's complete data payload can have a maximum length of 58 (for example see section 2.1.2 How to Send Commands On the USB Connection), so this field can have a maximum length of $58 - 6 = 52$ bytes.

Result Codes:

See command description.

Example Request (Hex)

Cmd Num	Data Len	Data
49	06	00 00 03 0D 00 00 [Extended Command 0x030D - Read Date and Time]

Example Response (Hex)

Result Code	Data Len	Data
0A	0D	00 00 00 00 00 07 06 14 11 00 00 00 01 (6/20/2009 5:00pm)

8.3.16 Command 0x4A - Get Extended Response (Extended Commands Only)

The host uses this command to retrieve additional response data longer than the current connection type's maximum packet size. After calling a command, if the device returns generic result code **0x0A Extended response first packet** (see **Table 8-3 - Generic Result Codes** on page 71), the host software should begin buffering the complete Extended Response starting with the initial response, then call this command repeatedly until it has retrieved the complete Extended Response.

The response data from the device follows the same Extended Data Offset rule as **Command 0x49 - Send Extended Command Packet** from the host: The first packet the device sends to the host specifies Extended Data Offset = 0, and subsequent packets, if any, specify Extended Data Offset = 52 (or other packet length depending on connection type), then 104, 156, and so on, until the device has sent all the response data. The host should continue sending this command to the device and buffering the returned Extended Data until the Extended Data Offset plus the length of the Extended Data equals the Complete Extended Data Length.

Request Data: None

Table 8-34 - Response Data for Command 0x4A - Get Extended Response (Extended Commands Only)

Offset	Field Name	Description
0..1	Extended Data Offset	This field is in big endian format. It indicates the byte offset position of this packet's Extended Data field, relative to the complete extended data field being sent as multiple packets. The first byte is offset zero.
2..3	Extended Result Code	This field is in big endian format. For one byte result codes, the high byte is set to zero.
4..5	Complete Extended Data Length	This field is in big endian format and gives the total length of the extended data field the device is returning to the host in multiple packets. If the complete extended data fits in a single packet, this field is equal to the Data Length field minus 6.
6..n	Extended Data	This field contains either part or all of the extended data the device is sending to the host. The size of this Extended Data field can be determined by subtracting the Extended Data field's offset within the response (6) from the response's total data length (N). In most cases the response's complete data payload can have a maximum value of 58 (for example see section 2.1.2 How to Send Commands On the USB Connection), so this field can have a maximum length of $58 - 6 = 52$ bytes.

Result Codes: Same as defined in **Command 0x49 - Send Extended Command Packet**.

Example Request (Hex)

Cmd Num	Data Len	Data
4A	00	

Example Response (Hex)

Result Code	Data Len	Data
0A	09	00 34 00 00 00 37 35 36 37 (Last 3 bytes of extended data out of 55 bytes)

8.3.17 Command 0x4C - Get Tamper Status (Tamper Only)

This command retrieves two bytes representing the device's tamper history and current tamper status, including which tamper circuits are active / armed to detect tampers, which ones have detected a tamper in the past, and which ones are currently registering a tamper. It also reports whether the device signature has been erased in response to tampering or removal of the tamper detection battery. When a device has detected a tamper, all incoming commands, including this one, fail.

Request Data: None

Table 8-35 - Response Data for Command 0x4C - Get Tamper Status (Tamper Only)

Offset	Field Name	Description
Byte 0	Tamper History	Bit 0 Tamper Armed Status: 1 = Tamper is active / armed 0 = Tamper is not active / armed Bit 1 Tamper Circuit 1 History: 1 = Circuit 1 was tampered 0 = Circuit 1 was not tampered Bit 2 Tamper Circuit 2 History: 1 = Circuit 2 was tampered 0 = Circuit 2 was not tampered Bit 3 Device Signature Tamper History: 1 = Device Signature was tampered 0 = Device Signature was not tampered Bit 4 Device Signature Erased History: 1 = Device Signature was erased 0 = Device Signature was not erased.
Byte 1	Tamper Status	Bit 0 Tamper Circuit 1 Status 1 = Circuit 1 is open 0 = Circuit 1 is closed Bit 1 Tamper Circuit 2 Status 1 = Circuit 2 is open 0 = Circuit 2 is closed

Result codes:
 0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
4C	00	

In the following example, tamper is armed, circuit 1 has registered a tamper, and the device signature was erased.

8 - Commands

Example Response (Hex)

Result Code	Data Len	Data
00	02	13 02

8.3.18 Command 0x4D - Configure General Status LED (PM3 Only)

Caution: Leaving the General Status LED continuously turned on when the device is running on battery power drastically reduces the battery life.

This command temporarily sets whether the General Status LED is Off or Solid Green when the device is battery powered and ready to read a card. This setting reverts to its default value when the device is power cycled or reset.

If the host doesn't specify an LED state value in the request, the device responds with the current LED state, otherwise the response contains no additional data.

Table 8-36 - Request Data for Command 0x4D - Configure General Status LED (PM3 Only)

Offset	Field Name	Description
0	LED State	0 = Off 1 = Green

Table 8-37 - Response Data for Command 0x4D - Configure General Status LED (PM3 Only)

Offset	Field Name	Description
0	LED State	0 = Off 1 = Green

Result codes: 0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
4D	01	01 (Green)

Example Response (Hex)

Result Code	Data Len	Data
00	00	

8.4 Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)

When calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, a value of 0x03 in the most significant byte of the Extended Command Number is reserved for EMV L2 commands, which are documented in this section.

8.4.1 About MACs

Many commands in this command group require a MAC field, which the host must populate using the UIK loaded into the device. For details, see section **4.1 About Message Authentication Codes (MAC)**.

8.4.2 About EMV L2 Transaction Flows (EMV Only)

The general flow of an EMV L2 transaction is as follows (bear in mind the device does not have a display, so in these steps the host drives the user interface for both the terminal operator / cashier and for the cardholder / customer):

- 1) The terminal operator / cashier performs steps external to the transaction, generally resulting in a total balance owed, and directs the host software to initiate a transaction. If the device supports Quick Chip and the system is designed to use that feature, the host may skip this step and instead start the transaction with a default amount as a placeholder, which is generally a pre-determined non-zero value that is consistent with the system's payment processing environment. Further differences pertaining to Quick Chip transactions are included in the steps below.
- 2) The host software sends the device **Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)**. If the host is using Quick Chip, it must specify Quick Chip operation in the **Options** field.
- 3) From this point until the host sends the device transaction results to the transaction processor, the host may cancel the EMV transaction by sending **Extended Command 0x0304 - Cancel Transaction (EMV Only)** and the device sends report **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction / Host Canceled EMV Transaction Before Card Was Presented**.
- 4) If the cardholder has not already presented payment, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Waiting for Cardholder Response / Waiting for Cardholder to Present Payment**, followed by **Notification 0x0301 - Display Message Request** to prompt the cardholder to **PRESENT CARD**. The device waits until the cardholder presents payment, pending a timeout.
- 5) Upon chip card insertion or contactless tap, the device sends **Notification 0x0300 - Transaction Status / Progress Information** to report **Card Inserted (or Contactless Token Detected) / Powering Up Card**.
- 6) (MSR Only) At this point, if the MSR is enabled and **Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)** indicated the MSR should be armed, the cardholder may swipe a magnetic stripe card:
 - a) On devices that do not support EMV MSR Flow (see **Table 1-2**), the device cancels the EMV transaction the host initiated, and reverts to the behavior described in section **6 Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)**.
- 7) (Contact Only) If the cardholder has inserted a chip card, the device attempts to communicate with the card. If it is unable to do so:
 - a) (Contact Only) On devices that do not support EMV MSR Flow (see **Table 1-2**), the device immediately terminates the transaction with no retries and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Payment Method Communication Error / Transaction Error**, followed by **Notification 0x0301 - Display Message Request** to the host with the message **TRANSACTION TERMINATED**.
- 8) The device negotiates with the card to determine which payment application to use as follows:

- a) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Selecting the Application**, followed by **Notification 0x0301 - Display Message Request** to prompt the cardholder to **PLEASE WAIT**.
 - b) If the card holds only one mutually supported payment application, the device proceeds to use that application. If the card holds more than one mutually supported application:
 - i) The device sends the host **Notification 0x0302 - Cardholder Selection Request** to prompt the cardholder to **Select Application** with a list of available applications, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report **Waiting for Cardholder Response / Waiting for Cardholder Application Selection**.
 - ii) After the cardholder selects an application, the host passes the selection to the device by sending **Extended Command 0x0302 - Cardholder Selection Result**.
 - c) The device sends **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Initiating Application**.
- 9) (Contact Only) If the cardholder has inserted a chip card, and the card's selected application reports to the device that the cardholder should select a language, the device sends the host **Notification 0x0302 - Cardholder Selection Request** to prompt the cardholder to **Select Language** with a list of available languages, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report event **Waiting for Cardholder Response / Waiting for Cardholder Language Selection**. After the cardholder selects a language, the host passes the selection to the device by sending **Extended Command 0x0302 - Cardholder Selection Result**.
- 10) The device initiates communication with the card and sends the host **Notification 0x0300 - Transaction Status / Progress Information** reporting **Transaction Progress Change/ Reading Application Data**. If an error or other type of failure occurs during this step, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0301 - Display Message Request** to the host with the message **TRANSACTION TERMINATED**, followed by **Notification 0x0304 - Transaction Result Message**.
- 11) Depending on the capabilities of the card and the device, the device authenticates the card data using SDA, DDA, or CDA. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Offline Data Authentication**.
- 12) The steps from here through **Card Action Analysis** below are collectively referred to as the **Risk Management** process.
- 13) The device checks to make sure the selected application is valid for the transaction, and is compatible with the device (such as application version number, application usage control, and application effective / expiration date), and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Process Restrictions**.
- 14) The device uses the cardholder verification related data in the card or contactless payment device to determine which cardholder verification method (CVMs) to use. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Cardholder Verification**.
- 15) The device performs terminal risk management procedures, which involves floor limit checking, velocity checking, and periodically forcing online authorization to protect against fraud, and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Terminal Risk Management**.
- 16) The device analyzes the results of the previous steps and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Terminal Action Analysis**.
- 17) The device rolls up the results of the previous Risk Management process:

- a) If the Risk Management process encounters an error or determines the transaction or payment method fails to meet required criteria, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0301 - Display Message Request** to the host with the message **TRANSACTION TERMINATED**, followed by **Notification 0x0304 - Transaction Result Message**, and terminates the transaction
 - b) If the Risk Management process determines the transaction is too risky to approve, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0304 - Transaction Result Message**, followed by **Notification 0x0301 - Display Message Request** with message **DECLINED** to notify the cardholder, and terminates the transaction.
- 18) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Generating First Application Cryptogram**. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** reporting **Transaction Progress Change/ Card Action Analysis**.
- 19) If the device is NOT configured as Online-Only Terminal Type [see **Appendix E EMV Terminal and Application Settings (EMV Only)**] and the Risk Management processes determined the transaction is OK to perform offline, the device reports the transaction result to the host as follows:
- a) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**.
 - b) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction** and either **Transaction Approved** or **Transaction Declined**, then sends the host **Notification 0x0301 - Display Message Request** with message **APPROVED** or **DECLINED** to notify the cardholder.
 - c) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.
- 20) If the device is configured as an Online Only Terminal Type [see **Appendix E EMV Terminal and Application Settings (EMV Only)**] or the Risk Management processes determined the transaction must be performed online, the device reports the transaction result to the host as follows:
- a) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Online Processing**, followed by **Notification 0x0303 - ARQC Message**.
 - b) The next event depends on whether the device supports the Contact Quick Chip feature or Contactless Quick Chip feature (see **Table 1-2**) and whether the host specified Quick Chip as an Option when it started the transaction:
 - c) If Quick Chip operation is supported and in effect:
 - i) The device immediately constructs its own internal ARPC Response, with tag 8A set to 'Z3' to coordinate the transaction with the card or other payment method, and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction / Transaction Declined**.
 - ii) The device sends the host **Notification 0x0301 - Display Message Request** with message **REMOVE CARD** to notify the cardholder the card can be removed.
 - iii) The host should then process the ARQC Message data, including replacing the default amount with the final transaction amount, and should coordinate with the transaction processor to retrieve a final transaction result. Because in this case the device is not involved

- in determining the final transaction result, it does not send a notification to the host to show **APPROVED** or **DECLINED**. Instead, the host should display an appropriate message (such as **QUICK CHIP APPROVED** / **QUICK CHIP DECLINED**) to the cardholder based on the final transaction result.
- iv) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.
 - d) If Quick Chip operation is NOT supported or is not in effect:
 - i) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Waiting for Online Processing Response**.
 - ii) The host processes the ARQC Message data and uses it to coordinate with the transaction processor to receive an ARPC Response, which it processes and sends to the device using **Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)**. (Contact Only) Alternatively, the host may implement host-driven Quick Chip by instead constructing its own preliminary ARPC Response with tag 8A set to 'Z3' and sending it to the device immediately, without waiting for a transaction processor response. The device responds by sending **Notification 0x0301 - Display Message Request** to the host with message **DECLINED** and ending the transaction. The host should suppress this message and take over the remainder of the transaction, including notifying the cardholder to remove the card, determining the final transaction amount, coordinating with the transaction processor to retrieve a final transaction result, and interacting with the cardholder.
 - iii) The device communicates with the chip card to determine whether to approve or decline the transaction, then sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**.
 - iv) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction** and either **Transaction Approved** or **Transaction Declined**, then and sends the host **Notification 0x0301 - Display Message Request** with message **APPROVED** or **DECLINED** to notify the cardholder of the transaction result.
 - v) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.

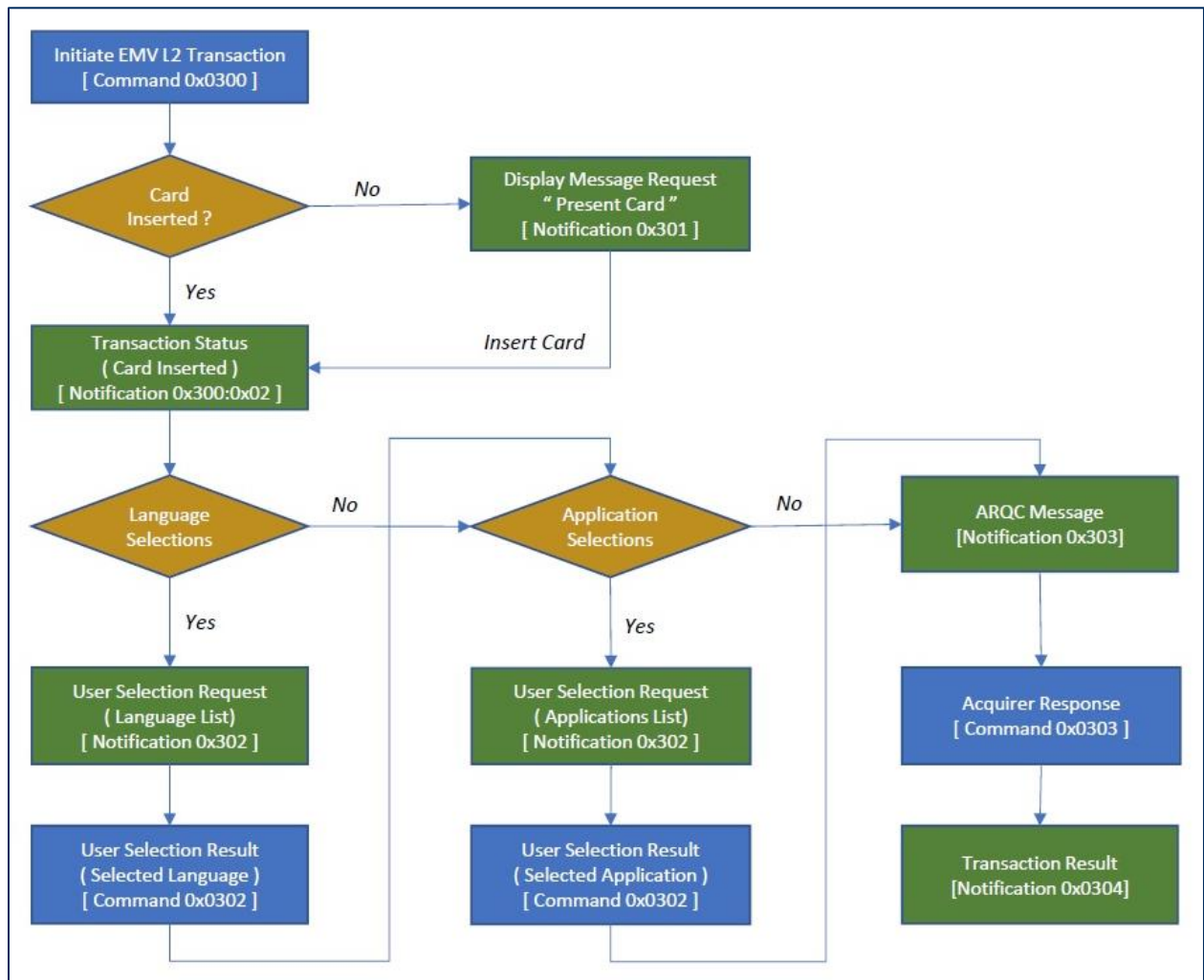


Figure 8-1 - Simplified EMV Transaction Flow

8.4.3 Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

CAUTION

After the host receives Notification 0x0304 - Transaction Result Message at the end of a transaction, it is very important to keep the device fully powered for at least 3 seconds. Disconnecting the device's power while it is advancing DUKPT keys after a transaction can corrupt the device's non-volatile memory and render the device unusable.

CAUTION

By default, when a transaction terminates without the cardholder presenting payment due to a timeout or Extended Command 0x0304 - Cancel Transaction (EMV Only), the device still sends Notification 0x0304 - Transaction Result Message and advances its DUKPT keys. In solution designs where this causes excessive consumption of DUKPT keys (such as solutions that continuously loop transactions to implement an "always armed" mode), the host should set Property 0x74 - EMV Transaction Result Format (EMV Only, Conserve DUKPT Keys Only) to Conserve DUKPT Keys.

The host uses this command to start an EMV transaction sequence, which flows as described in section **8.4.2 About EMV L2 Transaction Flows (EMV Only)**. The command provides all the data the device needs to start the transaction, and the device returns a response to the host to indicate whether the transaction will proceed. If the input fields for the command are not formatted correctly and within defined limits, the response message returns an error code indicating why the command could not proceed. If the device is set to a lower security level than **Security Level 3**, the device refuses this command, unless the device is an mDynamo, which accepts this command at **Security Level 2**.

If the command proceeds, the response indicates the transaction is proceeding. During transaction processing, the device may generate several notification messages. Some of these notifications may require the host to process data and initiate new commands. Whenever this happens, there is an associated timeout that causes the device to abandon the transaction with an error code if it occurs.

The device's system date and time must be set prior to sending this command:

- Devices that have a battery-backed real time clock (see **Table 1-2 - Device Features**) would typically have the date and time set at the factory.
- Devices without a battery-backed real time clock require the host to set the date and time using **Extended Command 0x030C - Set Date and Time (MAC)** every time the device is power cycled or reset.

After the host sends this command, the device is busy performing the EMV transaction. Until the transaction is complete or terminated, the host should only send commands to the device that directly pertain to the EMV transaction:

- **Extended Command 0x0302 - Cardholder Selection Result**
- **Extended Command 0x0303 - Online Processing Result / Acquirer Response**

- **Extended Command 0x0304 - Cancel Transaction (EMV Only)**
- **Extended Command 0x0305 - Modify Terminal Configuration (MAC)**

Table 8-38 - Request Data for Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)

Offset	Field Name	Value
0	Transaction Flow Time	<p>Specifies the maximum time, in seconds, for cardholder interaction events to complete while processing a transaction. Values from 0x01 to 0xFF are allowed (1 to 255 seconds).</p> <p>The timer starts at the beginning of each event. If the cardholder action does not occur within the specified time, the transaction proceeds as follows:</p> <ul style="list-style-type: none"> • Cardholder present payment timeout: The transaction terminates. • (Contact Only) Cardholder language selection timeout: The transaction continues with the default language. • (Contact Only) Cardholder application selection timeout: The transaction terminates.
1	Card Type to Read	<p>Card Type to Read (OR the following values together): 0x01 = Magnetic stripe card (MSR Only) 0x02 = Contact chip card (Contact Only)</p> <p>(MSR Only) Magnetic stripe card and Contact chip card can be enabled at the same time. For details about how the MSR and Contact functions interact, see the introduction to section 8.4 Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only).</p>
2	Options	<p>0x00 = Normal 0x01 = Reserved for Bypass PIN 0x02 = Reserved for Force Online</p> <p>(Quick Chip Only Contactless Quick Chip Only) To use Quick Chip mode, set the most significant bit to '1'. For example: 0x80 = Normal, Use Quick Chip</p>
3..8	Amount Authorized	Amount Authorized (EMV Tag 9F02, format n12, 6 bytes). For Transaction Type Refund (0x20), this must contain the refund amount.
9	Transaction Type	<p>0x00 = Purchase (covers transaction types Payment, Goods, and Services) 0x02 or 0x09 = Cash back (0x09 only supported when using contactless) 0x20 = Refund. If the specified Card Type to Read does not formally support refunds, the host can still use Refund to retrieve card data it needs to process a refund transaction, but internally and in its responses to the host, the device forces Transaction Type to Purchase and replaces Amount Authorized with 0.00.</p>
10..15	Cash Back	Cash back amount (if non-zero, EMV Tag 9F03, format n12, 6 bytes). For Transaction Type Refund (0x20) this must be 0.00.
16..17	Transaction Currency Code	<p>Transaction Currency Code (EMV Tag 5F2A, format n4, 2 bytes) Valid values are the numerical codes from <i>ISO 4217 Codes for the representation of currencies</i>, for example: 0x0000 = Use Selected Application's Currency Code Terminal Setting 0x0840 = US Dollar 0x0978 = Euro</p>

Offset	Field Name	Value
18	Reporting Option	This single byte field indicates the level of Transaction Status notifications the host wants the device to send during the transaction: 0x00 = Termination status only (normal termination, payment method communication or data error, timeout, host cancel) 0x01 = Major status changes (terminations plus card insertions and waiting for cardholder) (Contact Only) 0x02 = All status changes (documents the entire transaction flow)

Response Data: None. The response to this command only contains a result code.

Result codes:

- 0x0000 = Success, the transaction process has been started
- 0x0381 = Failure, DUKPT scheme is not loaded
- 0x0382 = Failure, DUKPT scheme is loaded but all of its keys have been used
- 0x0383 = Failure, DUKPT scheme is not loaded (Security Level not 3 or 4)
- 0x0384 = Invalid Total Transaction Time field
- 0x0385 = Invalid Card Type field
- 0x0386 = Invalid Options field
- 0x0387 = Invalid Amount Authorized field
- 0x0388 = Invalid Transaction Type field
- 0x0389 = Invalid Cash Back field
- 0x038A = Invalid Transaction Currency Code field
- 0x038E = Invalid Reporting Option
- 0x038F = Transaction Already In Progress
- 0x0391 = Invalid Device Serial Number
- 0x0396 = Invalid System Date and Time

Example Request (Hex)

Header	
Command Number	49
Data Length	19
Data	
Extended Data Offset	0000
Extended Command Number	0300
Complete Extended Data Length	0013
Extended Data	3C02000000000015000000000000000084002

Example Response (Hex)

Header	
Result Code	0A
Data Length	06

8 - Commands

Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.4 Extended Command 0x0302 - Cardholder Selection Result

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to respond to **Notification 0x0302 - Cardholder Selection Request**. After the device sends **Notification 0x0302 - Cardholder Selection Request** to the host, it expects the host to display the specified menu items to the cardholder, then, after the cardholder makes a selection, call **Extended Command 0x0302 - Cardholder Selection Result** to return the number of the item the cardholder selected. The number should be between 1 and the number of menu selection items being displayed. The first item, 0, is the title only.

Table 8-39 - Request Data for Extended Command 0x0302 - Cardholder Selection Result

Offset	Field Name	Value
0	Selection Status	<p>Indicates the status of Cardholder Selection:</p> <p>0x00 = Cardholder Selection Request completed, see Selection Result</p> <p>0x01 = Cardholder Selection Request cancelled by cardholder, Transaction Aborted</p> <p>0x02 = Cardholder Selection Request timed out, Transaction Aborted</p> <p>0x03 = Cardholder Selection Request timed out, Use Device Defaults (FEATURE TAG)</p> <p>The behavior of the device to each of the responses is dictated by EMV rules.</p>
1	Selection Result	Indicates the menu item selected by the cardholder. This is a single byte binary value.

Response Data: None. The response to this command only contains a result code.

Result codes:

0x0000 = Success, the Selection Result was received

0x038B = Invalid Selection Status

0x038C = Invalid Selection Result

0x038D = Failure, no transaction currently in progress

Example Request (Hex)

Header	
Command Number	49
Data Length	08
Data	
Extended Data Offset	0000
Extended Command Number	0302
Complete Extended Data Length	0002
Extended Data	0001

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.5 Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to inform the device of the result of on-line processing. It usually contains an ARPC and optionally Issuer Script 1 / Issuer Script 2 data.

Table 8-40 - Request Data for Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)

Offset	Field Name	Value
0	Message Length	Two byte binary, most significant byte first. This gives the total length of the ARPC message that follows, excluding padding and CBC-MAC.
2..n	Acquirer Response Message	This is the response from the acquirer. See Appendix D.2 ARPC Response from Online Processing for details.

Response Data: None. The response to this command only contains a result code.

Result codes:

0x0000 = Success, the Selection Result was received

0x038D = Failure, no transaction currently in progress

0x038F = Failure, transaction already in progress

Example Request (Hex)

Header	
Command Number	49
Data Length	39
Data	
Extended Data Offset	0000
Extended Command Number	0303
Complete Extended Data Length	003C
Extended Data	003AF92EDFDF540A00000000000000000DFDF550182DFD F250F423335453243443038303131364141FA0670048A0230300 0

Example Response (Hex)

Header	
Result Code	0B
Data Length	00
Data	
Extended Data Offset	Not Applicable

8 - Commands

Extended Result Code	Not Applicable
Complete Extended Data Length	Not Applicable
Extended Data	Not Applicable

Example Request Following Up For Packet 0 (Hex)

Header	
Result Code	49
Data Length	0F
Data	
Extended Data Offset	0033
Extended Result Code	0303
Complete Extended Data Length	003C
Extended Data	000000000000000000

Example Response Following Up For Packet 0 (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.6 Extended Command 0x0304 - Cancel Transaction (EMV Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to cancel a transaction while the device is waiting for the cardholder to present payment.

Request Data: None

Response Data: None

Result codes:

0x0000 = Success, the transaction was cancelled

0x038D = Failure, no transaction currently in progress

0x038F = Failure, transaction in progress, cardholder already presented payment

Example Request (Hex)

Header	
Command Number	49
Data Length	06
Data	
Extended Data Offset	0000
Extended Command Number	0304
Complete Extended Data Length	0000
Extended Data	Not Applicable

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.7 Extended Command 0x0305 - Modify Terminal Configuration (MAC)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command is used to directly modify tags in the device's EMV Terminal configuration. See **Extended Command 0x0306 - Read Terminal Configuration** and the Terminal Configuration subsections in **Appendix E EMV Terminal and Application Settings (EMV Only)**.

Some of the device's EMV Terminal configuration tags can only be set to EMV certified combinations. To change those settings, the host should use **Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)**.

Configuration changes will be lost after a power cycle or reset unless the host sends **Extended Command 0x030E - Commit Configuration** after making all configuration changes.

Table 8-41 - Request Data for Extended Command 0x0305 - Modify Terminal Configuration (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1	Slot Number	EMV Terminal Slot Number. Must be 0x01.
2	Operation	0x01 = Write Operation 0xFF = Set to Factory Defaults (sets all items, Terminal, Applications, and Application Public Keys to factory default values)
3	Database Selector	(Contact Only) 0x00 = EMV Contact L2
4..19	Device Serial Number (DSN)	16 Bytes DSN
20..n	Objects To Write	Note: Not needed if Operation is 0xFF Set to Factory Defaults. FA<len> /* container for generic data */ <tag><len><value> ... <tag><len><value>
n..n+3	MAC	MAC computed on Device Serial Number (DSN) and Objects to Write fields. See section 8.4.1 About MACs .

Response Data: None. The response to this command only contains a result code.

Result codes:

0x0000 = Success

0x0390 = Device Has No Keys

0x0391 = Invalid Device Serial Number

0x0392 = Invalid Type of MAC field

0x0393 = Invalid Slot Number field

0x0394 = Invalid Operation field

0x0395 = Invalid Database Selector field

8.4.8 Extended Command 0x0306 - Read Terminal Configuration

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to read EMV Terminal configuration data. See **Extended Command 0x0305 - Modify Terminal Configuration (MAC)** and the Terminal Configuration subsections in **Appendix E EMV Terminal and Application Settings (EMV Only)**.

Table 8-42 - Request Data for Extended Command 0x0306 - Read Terminal Configuration

Offset	Field Name	Value
0	Slot Number	EMV Terminal Slot Number. Must be 0x01.
1	Operation	0x00 = Read Operation 0x0F = Read All Tags of selected slot
2	Database Selector	(Contact Only) 0x00 = EMV Contact L2
3..	Tags to Read	Note: Not needed if Operation is 0x0F Read All Tags of selected slot. FA<len> /* container for generic data */ <tag> ... <tag> Tag DFDF47 cannot be read individually. This tag can only be retrieved using the 'Read All Tags' option.

Table 8-43 - Response Data for Extended Command 0x0306 - Read Terminal Configuration

Offset	Field Name	Value
0..1	Message Length	Two byte binary, most significant byte first. This gives the total length of the EMV Terminal Configuration message that follows.
2..	Tags Read	FA<len> /* container for generic data */ <tag><len><value> ... <tag><len><value> When reading all tags for the selected slot, the last two tags are: DFDF26, the Configuration Label DFDF47, the Database Checksum

Result codes:

0x0000 = Success

0x0393 = Invalid Slot Number field

0x0394 = Invalid Operation field

0x0395 = Invalid Database Selector field

0x0396 = Invalid Tags to Read field

Example Request (Hex)

Header	
Command Number	49
Data Length	0D
Data	
Extended Data Offset	0000
Extended Command Number	0306
Complete Extended Data Length	0007
Extended Data	010000FA029F1A

Example Response (Hex)

Header	
Result Code	0A
Data Length	11
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	000B
Extended Data	0009FA8200059F1A020840

8.4.9 Extended Command 0x0307 - Modify Application Configuration (MAC)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to modify EMV Application configurations. See **Extended Command 0x0308 - Read Application Configuration** and the Application Settings subsections in **Appendix E EMV Terminal and Application Settings (EMV Only)**.

Configuration changes will be lost after a power cycle or reset unless the host sends **Extended Command 0x030E - Commit Configuration** after making all configuration changes.

Table 8-44 - Request Data for Extended Command 0x0307 - Modify Application Configuration (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1	Slot Number	EMV Application Slot Number See Appendix E EMV Terminal and Application Settings (EMV Only) to determine how many application slots the device has for the selected database.
2	Operation	0x01 = Write Operation
3	Database Selector	(Contact Only) 0x00 = EMV Contact L2
4..19	Device Serial Number (DSN)	16 Bytes DSN
20..n	Objects to Write	FA<len> /* container for generic data */ <tag><len><value> ... <tag><len><value>
n..n+3	MAC	MAC computed on Device Serial Number (DSN) and Objects to Write fields. See section 8.4.1 About MACs .

Response Data: None. The response to this command only contains a result code.

Result codes:

- 0x0000 = Success
- 0x0390 = Device Has No Keys
- 0x0391 = Invalid Device Serial Number
- 0x0392 = Invalid Type of MAC field
- 0x0393 = Invalid Slot Number field
- 0x0394 = Invalid Operation field
- 0x0395 = Invalid Database Selector field
- 0x0396 = Invalid Objects to Write field
- 0x0397 = Invalid MAC

8.4.10 Extended Command 0x0308 - Read Application Configuration

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to read back EMV Application configurations. See **Extended Command 0x0307 - Modify Application Configuration (MAC)** and Appendix E.2.2 EMV Contact Application Settings.

Table 8-45 - Request Data for Extended Command 0x0308 - Read Application Configuration

Offset	Field Name	Value
0	Slot Number	EMV Application Slot Number See Appendix E EMV Terminal and Application Settings (EMV Only) to determine how many application slots the device has for the selected database.
1	Operation	0x00 = Read Operation 0x0F = Read All Tags of selected slot
2	Database Selector	(Contact Only) 0x00 = EMV Contact L2
3..	Tags to Read	Note: Not needed if Operation is 0x0F Read All Tags of selected slot. FA<len> /* container for generic data */ <tag> ... <tag>

Table 8-46 - Response Data for Extended Command 0x0308 - Read Application Configuration

Offset	Field Name	Value
0	Message Length	Two byte binary, most significant byte first. This gives the total length of the EMV Application Configuration message that follows.
2..	Tags Read	FA<len> /* container for generic data */ <tag><len><value> ... <tag><len><value>

Result codes:

0x0000 = Success

0x0393 = Invalid Slot Number field

0x0394 = Invalid Operation field

0x0395 = Invalid Database Selector field

0x0396 = Invalid Tags to Read field

Example Request (Hex)

Header	
Command Number	49
Data Length	0D

8 - Commands

Data	
Extended Data Offset	0000
Extended Command Number	0308
Complete Extended Data Length	0007
Extended Data	010000FA029F06

Example Response (Hex)

Header	
Result Code	0A
Data Length	15
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	000F
Extended Data	000DFA8200099F0606A00000002501

8.4.11 Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to modify CA Public Keys, which are specified by each of the payment brands and which the device can use to perform offline data authentication (ODA) to authenticate data from a chip card or contactless card or payment device on its own, in cases where network access to a payment processor is not available. See **Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)** for details about storage of keys.

Configuration changes will be lost after a power cycle or reset unless the host sends **Extended Command 0x030E - Commit Configuration** after making all configuration changes.

Table 8-47 - Request Data for Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = MSV5 MSCI CBC-MAC
1	Slot Number	CA Public Key Slot Number = Any value from 0x01 to 0x33 inclusive 0xFF = Next Available (slot with RID TLV length set to zero) If the Operation field is set to Erase All, this field is not used and can be set to any value.
2	Operation	0x00 = Erase All (Erases all tags in all CAPK slots). This sets the TLV length of every TLV data object in each slot to 1 and the value to 0. A slot is considered erased and available for use by the Next Available Slot Number (0xFF) if its RID TLV length is set to 1 and its value is set to 0. 0x01 = Writes a CA Public Key. To erase a single slot, write all of the slot's tags' TLV lengths to 1 and values to 0.
3	Database Selector	(Contact Only) 0x00 = EMV Contact L2
4..19	Device Serial Number (DSN)	16 Bytes DSN
20..n	Objects to Write	Note: Not needed if Operation is 0x00 Erase All. FA<len> /* container for generic data */ < DFDF79><len><value> /* RID */ < DFDF7A><len><value> /* Index */ < DFDF7B><len><value> /* Modulus */ < DFDF7C><len><value> /* Key Exponent */ < DFDF7D><len><value> /* Checksum */
n..n+3	MAC	MAC computed on Device Serial Number (DSN) and Objects to Write fields. See section 8.4.1 About MACs .

8 - Commands

Extended Data	Not Applicable
---------------	----------------

8.4.12 Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to read back CA Public Keys. For details about the purpose of these keys, see **Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)**.

Each CAPK database contains up to 51 key slots, each formatted as shown in **Table 8-49**.

Table 8-49 - Certificate Authority Public Key Slots 1 to 51 Data

Tag	Value (hex)	Length (bytes)	Max Length	Description
DFDF79	00	0x01	0x05	CA Public Key RID
DFDF7A	00	0x01	0x01	CA Public Key Index
DFDF7B	00	0x01	0xF8	CA Public key Modulus
DFDF7C	00	0x01	0x03	CA Public Key Exponent
DFDF7D	00	0x01	0x14	CA Public Key Checksum

Table 8-50 - Request Data for Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)

Offset	Field Name	Value
0	Slot Number	CA Public Key Slot Number = Any value from 0x01 to 0x33 inclusive
1	Operation	0x00 = Read Operation 0x0F = Read All Tags of selected slot
2	Database Selector	(Contact Only) 0x00 = EMV Contact L2
3..	Tags to Read	Note: Not needed if Operation is 0x0F Read All Tags of selected slot. FA<len> /* container for generic data */ <tag> ... <tag>

Table 8-51 - Request Data for Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)

Offset	Field Name	Value
0..1	Message Length	Two byte binary, most significant byte first. This gives the total length of the message that follows.
2..	Tags Read	FA<len> /* container for generic data */ <tag><len><value> ... <tag><len><value>

8 - Commands

Result codes:

0x0000 = Success

0x0393 = Invalid Slot Number field

0x0394 = Invalid Operation field

0x0395 = Invalid Database Selector field

0x0396 = Invalid Tags to Read field

Example Request (Hex)

Header	
Command Number	49
Data Length	09
Data	
Extended Data Offset	0000
Extended Command Number	030A
Complete Extended Data Length	0003
Extended Data	010F00

Example Response (Hex)

Header	
Result Code	0A
Data Length	25
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	001F
Extended Data	001DFA820019DFDF790100DFDF7A0100DFDF7B0100DFDF7C0100DFDF7D0100

8.4.13 Extended Command 0x030B - Read EMV Kernel Information

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to read kernel information.

Table 8-52 - Request Data for Extended Command 0x030B - Read EMV Kernel Information

Offset	Field Name	Value
0	Mode	(Contact Only) 0x01 = Version of EMV Contact L2 Kernel 0x11 = Checksum of EMV Contact L2 Kernel 0x12 = Checksum of EMV Contact L2 Configuration

Table 8-53 - Response Data for Extended Command 0x030B - Read EMV Kernel Information

Offset	Field Name	Value
0	Response Data	Requested kernel version or checksum. The kernel version is a human-readable string describing the kernel and its version, and the checksums are 40-character hexadecimal strings.

0x0000 = Success

0x0386 = Invalid Mode

Example Request (Hex)

Header	
Command Number	49
Data Length	07
Data	
Extended Data Offset	0000
Extended Command Number	030B
Complete Extended Data Length	0001
Extended Data	01

Example Response (Hex)

Header	
Result Code	0A
Data Length	1E
Data	
Extended Data Offset	0000
Extended Result Code	0000

8 - Commands

Complete Extended Data Length	0018
Extended Data	6544796E616D6F204C32204B65726E656C20526576204135 (eDynamo L2 Kernel Rev A5)

8.4.14 Extended Command 0x030C - Set Date and Time (MAC)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to set the device's date and time. See **Extended Command 0x030D - Read Date and Time**.

Devices with a battery-backed real time clock (see **Table 1-2 - Device Features**) have the date and time set by the manufacturer, so this command may not need to be used after that. Devices that do not have a battery-backed real time clock must use this command frequently because (a) the clock must be set before the device can process EMV transactions, and (b) the host software must use this command every time the device is power cycled or reset.

Table 8-54 - Request Data for Extended Command 0x030C - Set Date and Time (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1..16	Device Serial Number	16 Bytes Device Serial Number. Devices except the following can set this field to all zeroes. <ul style="list-style-type: none"> eDynamo with firmware part number 1000003354 revisions earlier than G02 eDynamo with firmware part number 1000002649
17	Month	Value from 0x01..0x0C
18	Day	Value from 0x01..0x1F (less depending on month)
19	Hour	Value from 0x00..0x17
20	Minute	Value from 0x00..0x3B
21	Second	Value from 0x00..0x3B
22	Unused	Value from 0x00..0x06
23	Year	Value from 0x00 (2008)..0x44 (2076)
24..27	MAC	MAC computed over all preceding fields except Type of MAC . Devices except the following can set this field to all zeroes. <ul style="list-style-type: none"> eDynamo with firmware part number 1000003354 revisions earlier than G02 eDynamo with firmware part number 1000002649

Response Data: None. The response to this command only contains a result code.

Result codes:

0x0000 = Success

0x0390 = Device Has No Keys

0x0391 = Invalid Device Serial Number

0x0392 = Invalid Type of MAC field

0x0396 = Invalid Date / Time data

0x0397 = Invalid MAC

8 - Commands

Example Request (Hex)

Header	
Command Number	49
Data Length	22
Data	
Extended Data Offset	0000
Extended Command Number	030C
Complete Extended Data Length	001C
Extended Data	000000000000000000000000000000021C0F380B0009xxxx xxxx Where xxxxxxxx is the 4-byte MAC

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.15 Extended Command 0x030D - Read Date and Time

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to get the date / time from the device's internal clock. See **Extended Command 0x030C - Set Date and Time (MAC)**.

Request Data: None

Table 8-55 - Response Data for Extended Command 0x030D - Read Date and Time

Offset	Field Name	Value
0	Month	Value from 0x01..0x0C
1	Day	Value from 0x01..0x1F (less depending on month)
2	Hour	Value from 0x00..0x17
3	Minute	Value from 0x00..0x3B
4	Second	Value from 0x00..0x3B
5	Unused	0x00
6	Year	Value from 0x00 (2008)..0xFF (2263)

Result codes:

0x0000 = Success

0x0396 = Invalid Date / Time data (Date / Time has not been set yet)

Example Request (Hex)

Header	
Command Number	49
Data Length	06
Data	
Extended Data Offset	0000
Extended Command Number	030D
Complete Extended Data Length	0000
Extended Data	Not Applicable

Example Response (Hex)

Header	
Result Code	0A
Data Length	0D
Data	
Extended Data Offset	0000

8 - Commands

Extended Result Code	0000
Complete Extended Data Length	0007
Extended Data	0204130D340009

8.4.16 Extended Command 0x030E - Commit Configuration

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

This command is used to commit configuration changes to non-volatile memory so they remain in place after a power cycle or reset. If this command is not sent after changing the configuration, the changes are lost on power cycle or reset.

Because non-volatile memory has limited erase/write cycles, the host should send this command after all configuration changes have been made. It should not be sent after each configuration change out of many.

Table 8-56 - Request Data for Extended Command 0x030E - Commit Configuration

Offset	Field Name	Value
0	Database Selector	(Contact Only) 0x00 = EMV Contact L2

Response Data: None

Result codes:

0x0000 = Success

0x0001 = Failure

0x0395 = Invalid Database Selector field

Example Request (Hex)

Header	
Command Number	49
Data Length	07
Data	
Extended Data Offset	0000
Extended Command Number	030E
Complete Extended Data Length	0001
Extended Data	00

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	0000
Extended Result Code	0000

8 - Commands

Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.17 Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to select a set of predetermined allowable values for the EMV configuration tags marked as only settable as part of Terminal Configuration in **Appendix E.2.1 EMV Contact Terminal Settings and Defaults (Contact Only)**. These values can not be set directly, they must be set to one of a specified set of values, selected from the list of **Vendor Config IDs** in the device's *Letter of Approval for Contact Level 2* posted in the list of *Approved / Evaluated* products on the EMVCo web site. Detailed descriptions of the tags set by this command can be found in *EMV Integrated Circuit Card Specifications for Payment Systems v4.3*.

Separate from these values, the host may set unrestricted tags directly using **Extended Command 0x0305 - Modify Terminal Configuration (MAC)** and **Extended Command 0x0307 - Modify Application Configuration (MAC)**.

Configuration changes will be lost after a power cycle or reset unless the host sends **Extended Command 0x030E - Commit Configuration** after making all configuration changes.

Table 8-57 - Request Data for Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1. (4 byte MAC)
1	Database Selector	0x00 = EMV Contact L2
2..17	Device Serial Number	16 Bytes DSN
18	Configuration ID	<p>One byte field that specifies one of the following configurations. Each device implements a subset of this standard list; the supported subset is specified in the device's EMVCo Letter of Approval (LoA) as Vendor Config IDs:</p> <p>0x00 = Vendor Config ID C1</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA enabled • No MSR Fallback • Signature, No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 set to 21 • Tag 9F33 set to 20 28 C8 • Tag 9F40 set to 72 00 00 B0 01 <p>0x01 = Vendor Config ID C2</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA disabled • No MSR Fallback, Signature

Offset	Field Name	Value
		<ul style="list-style-type: none"> • No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = 20 28 00 • Tag 9F40 = 72 00 00 B0 01 <p>0x02 = Vendor Config ID C3</p> <ul style="list-style-type: none"> • Attended, Offline/Online • SDA, DDA and CDA enabled • No MSR Fallback • Signature, No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 = 22 • Tag 9F33 = 20 28 C8 • Tag 9F40 = 72 00 00 B0 01 <p>0x03 = Vendor Config ID C4</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA enabled • With MSR Fallback • Signature, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit • Numeric, Alphabetic, Special, Command and Function keys • Print Attendant, Display Attendant, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = 60 28 C8 • Tag 9F40 = FF 80 F0 A0 01 <p>0x04 = Vendor Config ID C5</p> <ul style="list-style-type: none"> • Unattended, Online Only • SDA, DDA and CDA enabled • With MSR Fallback • No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Cardholder, Display Cardholder, Code Table 1 • Tag 9F35 = 24 • Tag 9F33 = 60 08 C8 • Tag 9F40 = FF 80 F0 50 01 <p>0x05 = Vendor Config ID C6</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA disabled

Offset	Field Name	Value
		<ul style="list-style-type: none"> • With MSR Fallback • Signature, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Attendant, Display Attendant, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = 60 28 00 • Tag 9F40 = FF 80 F0 A0 01 <p>0x06 = Vendor Config ID C7</p> <ul style="list-style-type: none"> • Unattended, Online Only • SDA, DDA and CDA disabled • With MSR Fallback • No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Cardholder, Display Cardholder, Code Table 1 • Tag 9F35 = 24 • Tag 9F33 = 60 08 00 • Tag 9F40 = FF 80 F0 50 01 <p>0x07 = Vendor Config ID C8</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA enabled • Manual Key Entry, With MSR Fallback • PIN, Signature, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Attendant, Display Attendant, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = E0 F8 C8 • Tag 9F40 = FF 80 F0 A0 01 <p>0x08 = Vendor Config ID C9</p> <ul style="list-style-type: none"> • Unattended, Online Only • SDA, DDA and CDA enabled • Manual Key Entry, With MSR Fallback • PIN, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Cardholder, Display Cardholder, Code Table 1 • Tag 9F35 = 24 • Tag 9F33 = E0 D8 C8

Offset	Field Name	Value
		<ul style="list-style-type: none"> • Tag 9F40 = FF 80 F0 50 01 <p>0x09 = Vendor Config ID C10</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA disabled • Manual Key Entry, With MSR Fallback • PIN, Signature, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Attendant, Display Attendant, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = E0 F8 00 • Tag 9F40 = FF 80 F0 A0 01 <p>0x0A = Vendor Config ID C11</p> <ul style="list-style-type: none"> • Unattended, Online Only • SDA, DDA and CDA disabled • Manual Key Entry, With MSR Fallback • PIN, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Cardholder, Display Cardholder, Code Table 1 • Tag 9F35 = 24 • Tag 9F33 = E0 D8 00 • Tag 9F40 = FF 80 F0 50 01 <p>0x0B = Vendor Config ID C12</p> <ul style="list-style-type: none"> • Attended, Online Only • SDA, DDA and CDA disabled • Manual Key Entry, With MSR Fallback • Signature, No CVM Required • Cash, Goods, Services, Cashback, Inquiry, Transfer, Payment, Administrative, Cash Deposit. • Numeric, Alphabetic, Special, Command and Function keys • Print Attendant, Display Attendant, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = E0 28 00 • Tag 9F40 = FF 80 F0 A0 01 <p>0x0C = Vendor Config ID C13</p> <ul style="list-style-type: none"> • Unattended, Online Only • SDA, DDA and CDA disabled • Manual Key Entry, With MSR Fallback • No CVM Required

8 - Commands

Extended Result Code	0000
Complete Extended Data Length	0000
Extended Data	Not Applicable

8.4.18 Extended Command 0x0311 - Read EMV Configuration (Contact Only)

Like all extended commands, the host initiates this command by calling **Command 0x49 - Send Extended Command Packet (Extended Commands Only)**, and receives a response as documented there.

The host uses this command to read which contact EMV configuration the device is using. For details, see **Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)**.

Table 8-58 - Request Data for Extended Command 0x0311 - Read EMV Configuration (Contact Only)

Offset	Field Name	Value
0	Database Selector	0x00 = EMV Contact L2

Table 8-59 - Response Data for Extended Command 0x0311 - Read EMV Configuration (Contact Only)

Offset	Field Name	Value
0	Configuration Identifier	One byte field containing the Configuration ID that was set using Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only) .

Result codes:

0x0000 = Success

0x0395 = Invalid Database Selector field

Example Request (Hex)

Header	
Command Number	49
Data Length	07
Data	
Extended Data Offset	0000
Extended Command Number	0311
Complete Extended Data Length	0001
Extended Data	00

Example Response (Hex)

Header	
Result Code	0A
Data Length	07
Data	
Extended Data Offset	0000
Extended Result Code	0000
Complete Extended Data Length	0001
Extended Data	01

9 Properties

9.1 About Properties

MagneSafe V5 devices have a number of programmable configuration properties stored in non-volatile memory. Most of the programmable properties pertain to data formats other than vendor-defined HID, but some of the properties deal with the device regardless of format (for information about changing formats and making format-specific properties visible, see **Property 0x10 - Interface Type**). These properties can be configured at the factory or by an administrator using software tools supplied by MagTek. Changing these configuration properties requires low-level communication with the device. Details for communicating with the device to read or change programmable properties are provided in section **8.3.1 Command 0x00 - Get Property** and section **8.3.2 Command 0x01 - Set Property (MAC)**.

9.2 Property 0x00 - Firmware ID

Property ID: 0x00
 Property Type: String
 Length: Fixed at 11 bytes
 Get Property: Yes
 Set Property: No
 Default Value: Part number of installed firmware

This is an 11 or 13 byte read-only property that identifies the firmware part number and revision installed on the device. The first 8 or 10 bytes represent the part number, the next byte represents the firmware major revision number, and the final two bytes represent an internal build number. For example, this property might be “21042812D01”.

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	00

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	0B	32 31 30 34 32 38 31 32 44 30 31

9.3 Property 0x01 - USB Serial Number (HID Only | KB Only)

Property ID: 0x01

Property Type: String

Length: 0 - 15 bytes

Get Property: Yes

Set Property: Yes

Default Value: Null string / ASCII device serial number set when the device is configured.

The value contains the USB serial number, from 0 to 15 bytes long. The device sends the value of this property (if any) to the host during USB device enumeration. This is useful for distinguishing between devices when more than one of the same kind of device is attached to the host.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	04	01	31 32 33

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	01

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	31 32 33

9.4 Property 0x02 - USB Polling Interval (HID Only | KB Only)

Property ID: 0x02
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x01

This one-byte value contains the device's polling interval in milliseconds for the **Interrupt In** Endpoint, can be between 1 - 255. The device sends the value of this property (if any) to the host during USB device enumeration, and the host can use it to determine how often to poll the device for USB Input Reports [see section **2.1.3 How to Receive Data On the USB Connection (HID Only)**]. For example, if the polling interval is set to 10, the host polls the device for Input Reports every 10ms. This property can be used to speed up or slow down the time it takes to send Input Reports to the host. The trade-off is that speeding up the polling interval increases the USB bus bandwidth used by the device.

If the USB host hardware is configured to use a small keyboard buffer, the device may drop characters and host software developers may use this setting to reduce the device's transmission speed to compensate. However, a better solution is to increase the host hardware's keyboard buffer size. For example, on a USB host with a buffer size of 100 bytes, increasing the buffer size to 1000 may allow much shorter polling intervals resulting in faster transmission speeds without reducing reliability. For details about adjusting keyboard buffer size, see the documentation about "Keyboard Buffer Size" for the specific host hardware.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	02	02	0A

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	02

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	01

9.5 Property 0x03 - Device Serial Number

Property ID: 0x03

Property Type: String

Length: 0 - 15 bytes

Get Property: Yes

Set Property: Yes (Once)

Default Value: Null string / ASCII device serial number set when the device is configured.

The property contains the device serial number, and is 0 to 15 bytes long. The device sends the value of this property (if any) to the host in the Device Serial Number field of **Magnetic Stripe Card Data Sent from Device to Host**, and in **ARQC Messages (EMV Only)**, **ARPC Response from Online Processing (EMV Only)**, and **Transaction Result Messages (EMV Only)**. This property may be Set only once; attempts to Set the property again fail with RC = 0x07 (Sequence Error). Note this value does not necessarily have the same value as **Property 0x01 - USB Serial Number (HID Only | KB Only)**, which is used mostly for differentiating identical devices after USB enumeration.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	04	03	31 32 33

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	03

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	31 32 33

9.6 Property 0x04 - MagneSafe Version Number

Property ID: 0x04
Property Type: String
Length: 0 - 7 bytes
Get Property: Yes
Set Property: No
Default Value: "V05"

This is a maximum 7-byte read-only property that identifies the MagneSafe Feature Level supported on this device. Attempts to set this property fail with RC = 0x01.

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	04

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	56 30 35

9.7 Property 0x05 - Track ID Enable (MSR Only)

Property ID: 0x05

Property Type: Byte

Length: 1 byte

Get Property: Yes

Set Property: Yes

Default Value: 0x95

This property is defined as follows:

Bit Position	7	6	5	4	3	2	1	0
	id	0	T ₃	T ₃	T ₂	T ₂	T ₁	T ₁

id = 0: Decodes standard ISO/ABA cards only

id = 1: Decodes AAMVA and 7-bit cards also

If the id flag is set to 0, only tracks that conform to the ISO card data format allowed for that track are decoded. If the track cannot be decoded by the ISO method, the device reports a decode error.

For each pair of track bits, valid values are as follows:

T_# = 00: Track Disabled

T_# = 01: Track Enabled

T_# = 10: Track Enabled and Required (Error if blank)

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	02	05	95

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	05

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	95

9.8 Property 0x07 - ISO Track Mask

Property ID: 0x07
 Property Type: String
 Length: 6 bytes
 Get Property: Yes
 Set Property: Yes
 Default Value: "04040Y"

This property specifies how the device should mask data on ISO/ABA type cards: Each byte in the sequence has the following meaning:

Offset	Description
0..1	These bytes are an ASCII representation of a decimal value that specifies how many of the leading characters of the PAN the device sends unmasked. The range is from "00" to "99".
2..3	These bytes are an ASCII representation of a decimal value that specifies how many of the trailing characters of the PAN the device sends unmasked. The range is from "00" to "99".
4	Masking Character. This byte specifies which character the device uses for masking. If this byte contains the uppercase letter 'V', the following rules apply: 1) The device masks the PAN using character '0' 2) The device leaves all data after the PAN unmasked, leaving Discretionary Data ("DD") and other non-PAN data available for the host to read.
5	This byte specifies whether the device applies Mod 10 Correction to the PAN. "Y" means Yes, "N" means No. This option is only effective if the Masking Character specified by this command is "0".

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

9.9 Property 0x08 - AAMVA Track Mask (MSR Only)

Property ID: 0x08
 Property Type: String
 Length: 6 bytes
 Get Property: Yes
 Set Property: Yes
 Default Value: "04040Y"

This property specifies the factors for masking data on AAMVA type cards. Each byte in the property has the following meaning:

Offset	Description
0..1	These bytes are an ASCII representation of a decimal value that specifies how many of the leading characters of the Driver's License/ID Number (DL/ID#) the device sends unmasked. The range is from "00" to "99".
2..3	These bytes are an ASCII representation of a decimal value that specifies how many of the trailing characters of the DL/ID# sends unmasked. The range is from "00" to "99".
4	<p>Masking Character. This byte specifies which character the device uses for masking. If this byte contains the uppercase letter 'V', the following rules apply:</p> <ul style="list-style-type: none"> • The device masks the PAN according to the rules of this property (Property 0x34 - Send AAMVA Card Data is ignored) • The device uses '0' for masking the PAN • The device sends all data after the PAN without masking
5	This byte specifies whether the device applies Mod 10 Correction to the DL/ID#. "Y" means Yes, "N" means No. This option is only effective if the masking character specified in this command is "0".

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

9.10 Property 0x0A - USB HID Max Packet Size (HID Only)

Property ID: 0x0A
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes (Read-Only on some devices)
 Default Value: 0x40

The value is a byte that contains the device's maximum packet size for the USB **Interrupt In** endpoint when using the HID data format [see section **2.1.3 How to Receive Data On the USB Connection (HID Only)**]. The device sends the value of this property to the host during USB device enumeration. The value can be set in the range of 1 - 64 and has units of bytes. For example, if the maximum packet size is set to 8, the device sends HID reports in multiple packets of 8 bytes each, possibly fewer bytes for the last packet of the report. This property can be used to speed up or slow down the time it takes to send data to the host. Larger packet sizes speed up communications and smaller packet sizes slow down communications. The trade-off is that speeding up the data transfer rate increases the USB bus bandwidth used by the device.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	02	0A	08

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	0A

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	08

9.11 Property 0x10 - Interface Type

Property ID: 0x10

Property Type: Byte

Length: 1 byte

Get Property: Yes

Set Property: Yes (No for devices that switch connections automatically)

Default Value: 0x00

This property represents the device's current connection type (see section **2 Connection Types**) and data format (see section **3 Data Formats**):

- Valid values for this property are
 - 0x00 = USB HID (HID Only)
- On devices that have only one possible value for this property, the property is read-only.
- On devices that support multiple values for this property and do not handle connection switching automatically, the host can use this property to change the device's behavior. MagTek strongly recommends the host set this property before setting other properties, and immediately power cycle or reset the device (see **Command 0x02 - Reset Device**), because it changes which other properties are available.
- (Bluetooth LE Only | iAP Only) This property only governs behavior of connections that are NOT Bluetooth LE or iAP (see section **1.4 About Connections and Data Formats**). Those connections are governed separately. The device uses the connection specified in **Property 0x5F - Notification Output Connection (Bluetooth LE Only | iAP Only, USB Only)** when it sends **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)** and **Notification Messages Sent from Device to Host (Extended Notifications Only)**.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Property Value
01	02	10	00

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	10

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	00

9.12 Property 0x15 - MagnePrint Flags (MSR Only)

Property ID: 0x15
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0x00

The host uses this property to direct the device to either include or exclude MagnePrint data in **Magnetic Stripe Card Data Sent from Device to Host (MSR Only | Keypad Entry Only)** when the device is in **Security Level 2**. At higher security levels, the device always sends encrypted MagnePrint data.

Bit Position	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	S

S = 0: Device does not include MagnePrint Data

S = 1: Device includes MagnePrint Data

Setting S to 1 directs the device to send **MagnePrint Status**, **MagnePrint Data Length (HID | GATT | SLIP)**, **MagnePrint Absolute Data Length (HID | TLV | GATT | SLIP)**, and **Encrypted MagnePrint Data** with each swipe when it is in **Security Level 2**.

Setting S to 0 directs the device to zero-fill these values.

Some devices are configured to only allow the manufacturer to modify this property.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

9.13 Property 0x31 - Mask Other Cards (MSR Only)

Property ID: 0x31
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0x00 (Don't Mask Other cards)

This property designates whether cards which do not decode as either ISO/ABA (Financial) or AAMVA (Driver License) format should be sent with their data masked or unmasked. The default value (0x00) is to send the data unmasked. If this property is set to 0x01, the device sends the track(s) to the host using a “0” for each byte of track data the device reads from the card.

If a card is encoded according to ISO/ABA rules (Track 1 in 7 bit format, Tracks 2 and Track 3 in 5 bit format), and Track 1 does not begin with the character ‘B’, the device always sends the **Track 1 Masked Data** value unmasked, regardless of the value of this property. See **Appendix E** for details.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

9.14 Property 0x33 - Card Inserted (MSR Insert Only | Contact Only)

Property ID: 0x33
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: No
Default Value: None

The host can use this read-only property to determine whether a card is fully inserted into the device. If a card is fully inserted, this equals 0x01, otherwise it equals 0x00.

For eDynamo, this property is available in firmware revisions 1000003354E00 (released June 2017) and later.

For mDynamo, this property is available in firmware revisions 1000003358C00 (released June 2017) and later.

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	33

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	01

9.15 Property 0x34 - Send AAMVA Card Data Unmasked (MSR Only)

Property ID: 0x34
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0x00

This property controls how the device sends AAMVA card data when the security level is higher than **Security Level 2**:

- 0 = Send masked AAMVA card data.
- 1 = Send clear AAMVA card data.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	34	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	34

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.16 Property 0x38 - HID SureSwipe Flag (SureSwipe Only, HID Only, MSR Only)

Property ID: 0x38
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x00

This property enables/disables SureSwipe emulation when in **Security Level 2** with **Property 0x10 - Interface Type** set to HID. This allows customers to receive a device without security enabled (**Security Level 2**) and use it in a similar manner to a SureSwipe device (for example, for convenience during software development). Later, when the customer is ready, they can switch the device to a higher Security Level and take advantage of the robust security features offered by the device.

When this property is set to 0x00, the device functions as described in this document.

When this property is set to 0x01, the device returns card swipes and enumerates with the same VID/PID as described in *D99875191 Technical Reference Manual, USB HID SureSwipe & Swipe Reader*. It does not emulate the property settings as defined there.

This property is only effective in USB HID mode.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	38	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	38

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.17 Property 0x52 - Host Poll Timeout (HID Only | KB Only)

Property ID: 0x52
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x02 (2 seconds)

The host can use this property to adjust the device's host poll timeout. The property can be set to 0 to disable the timeout, or it can be set to a value in the range of 1 to 60 seconds.

If the host fails to retrieve a USB HID input report from the device within the timeout period, the device discards the report. The intent of this timeout is to avoid having the device lock up while trying to send a report to a host that is failing to retrieve it due to error conditions or because the host is not ready to receive.

Not all devices have such a timeout, and not all readers implement this property to adjust it.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	52	02

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	52

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	02

9.18 Property 0x54 - Card Data Encryption Variant (MSR Only, Configurable MSR Variants Only)

Property ID: 0x54
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x00 (PIN Variant)

This property specifies which variant of the current DUKPT Key the device uses to encrypt magnetic stripe **Track 1 Encrypted Data**, **Track 2 Encrypted Data**, **Track 3 Encrypted Data**, and **Encrypted Session ID**:

- 0x00 = Use **PIN Encryption** variant
- 0x01 = Use **Data Encryption, request or both ways** variant

The host software should use this value to determine how to create the correct Derived Key to decrypt **Encrypted Track Data** (see section **5 Encryption, Decryption, and Key Management**). The algorithms for creating the Derived Key fitting each of the possible variants are fully specified in *ANS X9.24-1:2009*.

Some devices are configured to only allow the manufacturer to modify this property.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	54	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	54

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.19 Property 0x56 - MagnePrint Data Encryption Variant (MSR Only, Configurable MagnePrint Variants Only)

Property ID: 0x56
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x00 (PIN Variant)

This property specifies which variant of the current DUKPT Key the device uses to encrypt magnetic stripe **Encrypted MagnePrint Data**:

- 0x00 = Use **PIN Encryption** variant
- 0x01 = Use **Data Encryption, request or both ways** variant

The host software should use this value to determine how to create the correct Derived Key to decrypt **Encrypted MagnePrint Data** (see section 5 **Encryption, Decryption, and Key Management**). The algorithms for creating the Derived Key fitting each of the possible variants are fully specified in *ANS X9.24-1:2009*.

Some devices are configured to only allow the manufacturer to modify this property.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	56	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	56

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.20 Property 0x57 - SHA Hash Configuration (HID Only | TLV Only, Configurable SHA Only, MSR Only)

Property ID: 0x57
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x00

This property specifies whether and how the device returns a SHA-x Hash code with swipe data. See section **6.18 SHA-1 Hashed Track 2 Data (HID | TLV | GATT | SLIP, SHA-1 Only)**.

The possible options are:

Value	Meaning
0x00	Device sends a SHA-1 Hash code of all Track 2 data
0x01	Device sends a SHA-1 Hash code of the Track 2 PAN
0x02	Device sends a Salted SHA-1 Hash code of all Track 2 data
0x03	Device sends a Salted SHA-1 Hash code of the Track 2 PAN
0xFF	Device does not send any Hash code

Some devices are configured to only allow the manufacturer to modify this property.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	57	07

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	57

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.21 Property 0x5F - Notification Output Connection (Bluetooth LE Only | iAP Only, USB Only)

Property ID: 0x5F
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x01 (Bluetooth LE Only)

This property specifies which connection the device uses to send **Magnetic Stripe Card Data Sent from Device to Host** [see section 2 **Connection Types**] and **Notification Messages Sent from Device to Host (Extended Notifications Only)** to the host. To immediately and temporarily override the card swipe output connection, see **Command 0x48 - Notification Output Connection Override (Bluetooth LE Only | iAP Only, USB Only)**.

- 0x00 = USB connection
- 0x01 = Bluetooth LE connection (Bluetooth LE Only)

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	5F	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	5F

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.22 Property 0x67 - EMV Data Encryption Variant (EMV Only)

Property ID: 0x67
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x01 (Data Variant)

This property specifies which variant of the current DUKPT Key the device uses to encrypt EMV Data.

- 0x00 = Use the **PIN Encryption** variant.
- 0x01 = Use the **Data Encryption, request or both ways** variant.

The device uses this value to determine how to create the correct Derived Key to encrypt data involved in EMV transactions (see section **5 Encryption, Decryption, and Key Management**). The algorithms for creating the Derived Key fitting each of the possible variants are fully specified in *ANS X9.24-1:2009*.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	67	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	67

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

9.23 Property 0x6D - EMV Contact Notification Configuration (Contact Only)

Property ID: 0x6D
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x02 (Card Removed)

The host uses this property to enable or disable specific EMV notification messages the device sends with **Notification 0x0300 - Transaction Status / Progress Information**. Setting a bit to 1 enables the specified notification message, and setting a bit to 0 disables it.

The following table defines each bit in the property and describes which notification message it controls.

EMV Notification Enable: (Bit 0 is the least significant bit)

Bit	Field Name	Description
0	Card Inserted	When an EMV transaction is not in progress, this setting controls whether the device sends Notification 0x0300 - Transaction Status / Progress Information with its event field set to 0x01 = Card Inserted when a cardholder inserts a card into the EMV card slot. When an EMV transaction is in progress, insertion notifications are controlled using the Reporting Option field in Extended Command 0x0300 - Initiate EMV Transaction (EMV Only) .
1	Card Removed	Controls whether the device sends Notification 0x0300 - Transaction Status / Progress Information message with its event field set to 0x08 = Card Removed when the cardholder removes a card from the EMV card slot.
2-7	Reserved	Always set to zeroes.

For eDynamo, this property is available in firmware revisions 1000003354E00 (released June 2017) and later.

For mDynamo, this property is available in firmware revisions 1000003358C00 (released in June 2017) and later.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	6D	02

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

9 - Properties

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	6D

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	02

9.24 Property 0x74 - EMV Transaction Result Format (EMV Only, Conserve DUKPT Keys Only)

Property ID: 0x74
 Property Type: Byte
 Length: 1 byte
 Get Property: Yes
 Set Property: Yes
 Default Value: 0x00 (Legacy format)

The host uses this property to control the format of **Notification 0x0304 - Transaction Result Message**.

- **0x00 = Use Legacy Format.** The device sends the message in the standard format described in **Appendix D.3.2 Transaction Result Message Format Security Level 3**.
- **0x01 = Conserve DUKPT Keys.** The device omits the TLV data object F8 for encryption from **Transaction Result Message Format Security Level 3** if the transaction terminates without the cardholder presenting payment, which can occur if the host cancels the transaction or if the transaction times out. This prevents the device from advancing any DUKPT keys in this situation.

This property is stored in non-volatile memory, so it persists when the device is power cycled. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	74	00

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	74

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	00

Appendix A Bluetooth LE Controller Properties (Bluetooth LE Only)

The properties in the following subsections can be get and/or set using **Bluetooth LE Command 0x00 - Get Property** and **Bluetooth LE Command 0x01 - Set Property**.

A.1 Bluetooth LE Property 0x00 - Bluetooth LE Firmware ID

Bluetooth LE Property ID: 0x00

Get Property: Yes

Set Property: No

Default value: None

This is an 11 byte read-only property that identifies the firmware part number and revision for the firmware that resides in the device's Bluetooth LE controller. The first 8 bytes represent the firmware part number and the last 3 bytes represent the revision. For example, this property might be "21043029B04."

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 00

Example Get Response (Hex)

Result Code	Data Len	Data
00	0E	01 01 00 32 31 30 34 33 30 32 39 42 30 34 (value "21043029B04")

A.2 Bluetooth LE Property 0x01 - Bluetooth LE Device Address

Bluetooth LE Property ID: 0x01

Get Property: Yes

Set Property: No

Default value: None

This is a 6 byte read-only property that contains the Bluetooth LE device address. The first byte contains the least significant byte of the address. This address varies with each device.

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 01

Example Get Response (Hex)

Result Code	Data Len	Data
00	09	01 01 00 EC 11 A0 E5 C5 78 (value 0x78C5E5A011EC)

A.3 Bluetooth LE Property 0x02 - Bluetooth LE Device Name

Bluetooth LE Property ID: 0x02

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default Value: String <Product Name>-XXYY, where XX is the second-to-least significant byte of the Bluetooth LE device address converted to ASCII hex, and YY is the least significant byte. For example, if the second to least significant byte of an eDynamo’s Bluetooth LE device address is 0x11 and the least significant byte is 0xEC, the Bluetooth LE device name would be eDynamo-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.

This property contains the Bluetooth LE device name, which the Bluetooth LE host typically uses to present the operator with a choice of devices to interact with. If more than one device of the same name is available, MagTek recommends including a unique identifier in the device name and labeling the device accordingly so to visually distinguish one device from another.

This property can be 0 to 20 ASCII characters long. It should not contain any null characters (0x00). If set to a length of 0, the value reverts to the original default value.

Changes made to this property persist even if the device is powered off or reset. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	07	01 00 01 02 31 32 33 (value “123”)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.4 Bluetooth LE Property 0x03 - Configuration Revision

Bluetooth LE Property ID: 0x03

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 0

This property is a one-byte value between 0 and 255 the host can use to track the device's configuration status. For example, the host may read the default value of 0 and determine the module needs to be configured, then configure the device and set the value to 1 to indicate configuration is complete. On subsequent powerups, the host could then verify the property equals 1 before proceeding with normal operation, or perform further configuration steps and advance the property to 2.

Changes made to this property persist even if the device is powered off or reset. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 03

Example Get Response (Hex)

Result Code	Data Len	Data
00	04	01 01 00 01 (value 1)

A.5 Bluetooth LE Property 0x07 - Passkey

Bluetooth LE Property ID: 0x07

Get Property: No

Set Property: Yes

Non-Volatile: Yes

Default value: 0 (representing passkey 000000)

This property is a four-byte integer that represents the six-decimal-digit Bluetooth LE passkey (for example, 123456). To maximize the security of the Bluetooth LE connection, the passkey should be changed to something other than its default value by an administrator. The minimum value of the property is decimal 000000, and the maximum value of the property is decimal 999999. The first byte is the least significant byte (LSB).

Changes made to this property persist even if the device is powered off or reset. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	08	01 00 01 07 3F 42 0F 00 (value 999999 decimal)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.6 Bluetooth LE Property 0x08 - Configuration Bits

Bluetooth LE Property ID: 0x08

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default Value: Depends on data format. See **Table 9-1**.

Table 9-1 - Configuration Bits Default Values Per Data Format

Bluetooth LE Interface Type	Default value	USB Power Not Exit Airplane Mode	Never Advertise	Normally Connectable	Use Whitelist
HID	0x02	False	False	True	False
KB	0x00	False	False	False	False
GATT	0x02	False	False	True	False

This property is a one byte value that contains configuration bits that control various Bluetooth LE features. Bits 7-2 are reserved for future use and should always be set to 0.

Bit Position	7	6	5	4	3	2	1	0
Decode Type	R	R	R	R	USB Power Not Exit Airplane Mode	Never Advertise	Normally Connectable	Use Whitelist

Bit 0 is the **Use Whitelist** bit. Not all devices allow the host to set this bit. When this bit is set, the device behaves according to Bluetooth LE standard whitelist rules, which prevents unpaired hosts that are not on the device’s whitelist from connecting to the device when it is advertising. This makes the device compliant with the HID over GATT profile defined by the Bluetooth LE standard. Setting this bit is appropriate only for solutions where the Bluetooth LE host has a fixed Bluetooth LE address; Bluetooth LE hosts that use random Bluetooth LE addresses – such as iPhones and other Apple devices – will fail to reconnect, because random Bluetooth LE addresses are incompatible with whitelisting.

Bit 1 is the **Normally Connectable** bit. Not all devices allow the host to set this bit. When this bit is set, the device always advertises if it is not connected to a Bluetooth LE host, even when it has no card data to send. Because the device’s advertising controls whether a Bluetooth LE host can connect, this flag effectively allows the host to connect at will. This setting should be considered carefully, because granting the Bluetooth LE host full control over the connection state can drain the device’s battery, but it can be useful in specific cases:

- If the host needs to send commands over Bluetooth LE at any time, or
- If the battery drain is worth eliminating any delays generally introduced by re-connecting every time the device has card data to send.

When the Normally Connectable bit is set to 1, it is usually also desirable to only have the host initiate Bluetooth LE disconnects, instead of the device. To prevent the device from disconnecting from the

Bluetooth LE host automatically, set **Bluetooth LE Property 0x0B - General Connection Timeout** to 0 (Disabled).

(Custom Advertising Only) Bit 2 is the **Never Advertise** bit. When this bit is set to 1, Bluetooth LE never advertises. This effectively disables Bluetooth LE functionality for solutions that only require use of other physical connection types, such as USB. On devices that do not support it, this bit is reserved and should always be written with zero.

(Custom Advertising Only) Bit 3 is the **USB Power Not Exit Airplane Mode** bit. By default, applying USB power to the device triggers it to exit airplane mode and start advertising. Setting this bit to 1 disables this behavior. Pressing and releasing the button can still be used to exit airplane mode, regardless of how this bit is set. On devices that do not support it, this bit is Reserved and should always be written with zero.

Bits 4 to 7 are reserved. These bits should always be written with zeroes.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Changes made to this property persist even if the device is powered off or reset. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 08 01 (use white list bit is set)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.7 Bluetooth LE Property 0x0B - General Connection Timeout

Bluetooth LE Property ID: 0x0B

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default Value: Depends on data format. See **Table 9-2**.

Table 9-2 - General Connection Timeout Property Default Values Per Data Format

Bluetooth LE Interface Type	Default value
HID	0 (disabled)
KB	20000 (milliseconds)
GATT	0 (disabled)

This property is a four byte integer in least significant byte order that sets how long the device stays connected to the Bluetooth LE host when there has been no communication. The device disconnects from the Bluetooth LE host after this timeout expires. This can be used for power saving purposes. In addition, devices that use the KB connection type may prevent some hosts from displaying their virtual touch keyboards when the device is connected, so disconnecting when not in use may be desirable.

Setting this property to zero stops the device from timeout-disconnecting from the Bluetooth LE host, which causes the battery to drain more quickly.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	08	01 00 01 0B 20 4E 00 00 (20000 (0x4E20) milliseconds)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.8 Bluetooth LE Property 0x0C - Desired Minimum Connection Interval

Bluetooth LE Property ID: 0x0C

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 10 (12.5 milliseconds)

This property is a two byte integer in least significant byte order, in 1.25 millisecond increments, that contains the **Interval Min** value the device sends to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST (see the core Bluetooth specification for details). Only values between 6 and 3200 are valid.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 0C 0A 00 (12.5 milliseconds)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.9 Bluetooth LE Property 0x0D - Desired Maximum Connection Interval

Bluetooth LE Property ID: 0x0D

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 10 (12.5 milliseconds)

This property is a two byte integer in least significant byte order, in 1.25 millisecond increments, that contains the **Interval Max** value the device sends to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST (see the core Bluetooth specification for details). Only values between 6 and 3200 are valid.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 0D 0A 00 (12.5 milliseconds)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.10 Bluetooth LE Property 0x0E - Desired Slave Latency

Property identifier: 0x0E

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 4

This property is a two byte integer in least significant byte order that contains the **Slave Latency** value the device sends to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST (see the core Bluetooth specification for details). Only values between 0 and 499 are valid.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 0E 04 00 (value 4)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.11 Bluetooth LE Property 0x0F - Desired Supervision Timeout

Bluetooth LE Property ID: 0x0F

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 500 (5000 milliseconds)

This property is a two byte integer in least significant byte order, in 10 millisecond increments, that contains the **Timeout Multiplier** value the device sends to the Bluetooth LE host in a CONNECTION PARAMETER UPDATE REQUEST (see the core Bluetooth specification for details). Only values between 10 and 3200 are valid.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off for at least 30 seconds, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 0F F4 01 (5000 milliseconds)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.12 Bluetooth LE Property 0x12 - Connection Parameter Update Request Control

Bluetooth LE Property ID: 0x12

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

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

This property is a one byte value whose bits control various connection parameter update features. Bits 7-1 are reserved for future use and should always be set to 0.

Bit 0 = **Send Connection Parameter Update** bit. When this bit is set to 1, the device sends a connection parameter update request once after each Bluetooth LE connection.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 12 01 (send connection parameter update bit is set)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.13 Bluetooth LE Property 0x13 - Bluetooth Status LED Functionality Control (Pairing Modes Only)

Bluetooth LE Property ID: 0x13

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 0x00 (Off During Bluetooth LE Connection)

This property is a one byte value that controls the Bluetooth Status LED functionality. On devices that do not have a dedicated Bluetooth Status LED, the device always returns 0x00.

When this byte is set to 0x00, the Bluetooth Status LED is OFF when the device is connected to a host via Bluetooth LE, which saves battery power.

When this byte is set to 0x01, the Bluetooth Status LED is ON when the host has established an *encrypted* Bluetooth LE connection with the device, indicating the device is accepting commands and transactions. This provides additional visual cues for cardholders and operators, but uses more battery power.

When this byte is set to 0x02, the Bluetooth Status LED is ON when the host has established *any* Bluetooth LE connection with the device. If the connection is not yet encrypted, the device does not process commands or transactions. This option can be useful for diagnosing whether a host is connected to the device. When a host is connected to the device, the device does not advertise and is not able to connect to any other host until the connection is broken.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 13 00 (off during Bluetooth LE connection)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.14 Bluetooth LE Property 0x15 - Pairable Timeout (Pairing Modes Only)

Bluetooth LE Property ID: 0x15

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 0x03 (3 minutes)

This property is a one byte value that controls how many minutes the device waits to be paired before exiting pairing mode. This property can be set to a value between 0 and 5 minutes. When set to 0, the device is always pairable if it is not in Airplane Mode.

When set to 0, the Bluetooth Status LED blinks on briefly every 2 seconds for one minute after the device exits airplane mode to indicate the device is pairable. After one minute, the LED turns off to conserve power, but the device remains pairable.

When set to a value between 1 and 5 minutes and when the device is not pairable, the device rejects pairing requests from any host that tries to pair with it. To make the device pairable, press the button for two seconds until the Bluetooth Status LED changes from solid on to blinking, then immediately release the button. Do not hold the button for more than one second past the three flashes, or the device resets and is not pairable. While the device is pairable, the LED blinks on briefly every 2 seconds for the number of minutes the device is pairable for.

On devices that do not support this property, the firmware behaves as if the property is set to 0. On devices that do support it, the default is now set to 3, and the firmware behavior changes accordingly when using the new default.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 15 03 (03 = 3 minutes)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.15 Bluetooth LE Property 0x16 - Maximum Bond Count (Pairing Modes Only)

Bluetooth LE Property ID: 0x16

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 0x09 (9 bonds)

This property is a one byte value that controls how many hosts the device remains bonded with at one time. The property can be set to a value between 1 and 9 bonds. See **Bluetooth LE Property 0x17 - Maximum Bond Mode (Pairing Modes Only)** for a description of how the device behaves when the maximum number of bonds is reached.

Changing this property automatically causes the device to erase all bonds. The operator should unpair and then re-pair with any host that was previously paired with the device before trying to connect to the device with that host.

When this property is changed, the device must be reset manually or with a command (see **Command 0x02 - Reset Device**), before using the device further with a Bluetooth LE connection. Because this property affects Bluetooth LE communication, it is best to change it using the USB connection.

On devices that do not support this property, the firmware behaves as if this property is set to 9.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 16 09 (09 = 9 bonds)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.16 Bluetooth LE Property 0x17 - Maximum Bond Mode (Pairing Modes Only)

Bluetooth LE Property ID: 0x17

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 0x00 (FIFO mode)

This property is a one byte value that controls how the device behaves when the maximum number of bonds [controlled by **Bluetooth LE Property 0x16 - Maximum Bond Count (Pairing Modes Only)**] is reached. This property can be set to value 0 (FIFO) or 1 (Not Pairable).

When the property is set to 0 (FIFO) and an operator attempts to pair with a new host when the device is bonded with the maximum number of Bluetooth LE hosts, the device deletes the oldest bond and continues to pair/bond with the new host.

When the property is set to 1 (Not Pairable) and the device is bonded with the maximum number of Bluetooth LE hosts, the device leaves pairing mode and can not be placed into pairing mode until all bonds have been erased using **Bluetooth LE Command 0x07 - Erase All Bonds**.

When this property is changed, the device must be reset manually or with a command (see **Command 0x02 - Reset Device**), before using the device further with a Bluetooth LE connection. Because this property affects Bluetooth LE communication, it is best to change it using the USB connection.

On devices that do not support this property, the firmware behaves as if this property is set to 0.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	05	01 00 01 17 00 (00 = FIFO)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

A.17 Bluetooth LE Property 0x18 - Minimum Background Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x18

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **minimum background advertising interval**. This property, combined with **Bluetooth LE Property 0x19 - Maximum Background Advertising Interval (Custom Advertising Only)**, determines the Bluetooth LE advertising interval the device uses when the initial or pairing advertising interval is not in effect. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern when running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior. **Bluetooth LE Property 0x19 - Maximum Background Advertising Interval (Custom Advertising Only)** may also need to be adjusted when changing this property. If the maximum background advertising interval is less than the minimum, the device may behave unpredictably.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, MagTek recommends only changing it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 18 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 18

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

A.18 Bluetooth LE Property 0x19 - Maximum Background Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x19

Get Property: Yes

Set Property: Yes

Non-Volatile: Yes

Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **maximum background advertising interval**. This property, combined with **Bluetooth LE Property 0x18 - Minimum Background Advertising Interval (Custom Advertising Only)**, determines the Bluetooth LE advertising interval the device uses when no other advertising interval is in effect. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern when running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior.

Bluetooth LE Property 0x18 - Minimum Background Advertising Interval (Custom Advertising Only) may also need to be adjusted when changing this property. Using a maximum background advertising interval less than the minimum causes unpredictable behavior.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, it is recommended to only change it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 19 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 19

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

A.19 Bluetooth LE Property 0x1C - Minimum Initial Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x1C
 Get Property: Yes
 Set Property: Yes
 Non-Volatile: Yes
 Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **minimum initial advertising interval**. This property, combined with **Bluetooth LE Property 0x19 - Maximum Background Advertising Interval (Custom Advertising Only)**, determines the Bluetooth LE advertising interval the device uses for one minute after the device exits airplane mode. MagTek recommends setting both properties to the same value. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern when running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior. **Bluetooth LE Property 0x1D - Maximum Initial Advertising Interval (Custom Advertising Only)** may also need to be adjusted when changing this property. If the maximum initial advertising interval is less than the minimum, the device may behave unpredictably.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, it is recommended to only change it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 1C 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 1C

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

A.20 Bluetooth LE Property 0x1D - Maximum Initial Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x1D
 Get Property: Yes
 Set Property: Yes
 Non-Volatile: Yes
 Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **maximum initial advertising interval**. This property, combined with **Bluetooth LE Property 0x1C - Minimum Initial Advertising Interval (Custom Advertising Only)**, determines the Bluetooth LE advertising interval the device uses for one minute after the device exits airplane mode. MagTek recommends setting both properties to the same value. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern when running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior. **Bluetooth LE Property 0x1C - Minimum Initial Advertising Interval (Custom Advertising Only)** may also need to be adjusted when changing this property. If the maximum initial advertising interval is less than the minimum, the device may behave unpredictably.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, it is recommended to only change it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 1D 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 1D

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

A.21 Bluetooth LE Property 0x1E - Minimum Pairable Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x1E
 Get Property: Yes
 Set Property: Yes
 Non-Volatile: Yes
 Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **minimum pairable advertising interval**. This property, combined with **Bluetooth LE Property 0x1F - Maximum Pairable Advertising Interval (Custom Advertising Only)**, determines the advertising interval the device uses when it is pairable. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern if the device is running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior. **Bluetooth LE Property 0x1F - Maximum Pairable Advertising Interval (Custom Advertising Only)** may also need to be adjusted when changing this property. If the maximum pairable advertising interval is less than the minimum, the device may behave unpredictably.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, it is recommended to only change it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 1E 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 1E

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

A.22 Bluetooth LE Property 0x1F - Maximum Pairable Advertising Interval (Custom Advertising Only)

Bluetooth LE Property ID: 0x1F
 Get Property: Yes
 Set Property: Yes
 Non-Volatile: Yes
 Default value: 32 (20 milliseconds)

This property is a two byte integer in least significant byte order, in 625 microsecond increments, that contains the **maximum pairable advertising interval**. This property, combined with **Bluetooth LE Property 0x1E - Minimum Pairable Advertising Interval (Custom Advertising Only)**, determines the advertising interval the device uses when it is pairable. Smaller advertising intervals cause the device to consume more power when advertising, which may be a concern when running on battery power.

Only values between 32 (20ms) and 16384 (10.24s) are valid. Using values outside this range causes unpredictable behavior. **Bluetooth LE Property 0x1E - Minimum Pairable Advertising Interval (Custom Advertising Only)** may also need to be adjusted when changing this property. If the maximum pairable advertising interval is less than the minimum, the device may behave unpredictably.

This property is stored in non-volatile memory, so it persists when the device is power cycled. This property would typically only be changed once during device configuration if needed. Modifying this property too many times wears out flash memory.

Because this property affects Bluetooth LE behavior, it is recommended to only change it using the USB interface. When this property is changed, the device must be reset (see **Command 0x02 - Reset Device**) or powered off, then powered on, before the changes will take effect.

Example Set Request (Hex)

Cmd Num	Data Len	Data
46	06	01 00 01 1F 40 06 (1000ms / .625ms) = 1600 (0x0640)

Example Set Response (Hex)

Result Code	Data Len	Data
00	03	01 01 00

Example Get Request (Hex)

Cmd Num	Data Len	Data
46	04	01 00 00 1F

Example Get Response (Hex)

Result Code	Data Len	Data
00	05	01 01 00 40 06 (1000ms / .625ms) = 1600 (0x0640)

Appendix B Examples

This section includes direct command examples and information about using demonstration software. In addition to the examples here, source code with detailed comments is included with the demo software and can be used as a guide for custom software development.

The book *USB Complete* by Jan Axelson is also a good guide for software developers, especially the chapter “Human Interface Devices: Host Applications.”

B.1 Command Examples

This section provides examples of command sequences and cryptographic operations. Each example shows a sequence as it actually runs, so developers of custom software can check their code against the examples step-by-step to make sure the software is parsing and computing values correctly.

B.1.1 Example: HID Device Card Swipe In Security Level 2 (HID Only, MSR Only)

This example shows the data received in a HID report for a device at **Security Level 2** [see section 2.1 **How to Use USB Connections (USB Only)**].

The raw HID report is:

Byte	Content
0	00 00 00 3C 25 1F 00 25 42 35 34 35 32 33 30 30 35 35 31 32
20	32 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20 20
40	20 20 5E 30 38 30 34 33 32 31 30 30 30 30 30 30 30 37 32 35
60	30 30 30 30 30 30 3F 00 00 00 00 00 00 00 00 00 00 00 00 00
80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3B
120	35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 3D 30 38 30
140	34 33 32 31 30 30 30 30 30 30 30 37 32 35 30 3F 00 00 00 00
160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
220	00 00 00 00 00 00 00 00 00 00 00 3B 35 31 36 33 34 39 39 30
240	38 30 30 32 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30 30
260	30 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
360	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
380	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
440	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
460	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00
500	00 00 00 00 00 3C 25 1F 25 42 35 34 35 32 33 30 30 35 35 31
520	32 32 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20
540	20 20 20 5E 30 38 30 34 33 32 31 30 30 30 30 30 30 30 37 32
560	35 30 30 30 30 30 30 3F 00 00 00 00 00 00 00 00 00 00 00 00
580	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
600	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
620	3B 35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 3D 30 38
640	30 34 33 32 31 30 30 30 30 30 30 30 37 32 35 30 3F 00 00 00
660	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
680	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
700	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
720	00 00 00 00 00 00 00 00 00 00 00 00 00 3B 35 31 36 33 34 39 39
740	30 38 30 30 32 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30
760	30 30 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
780	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
800	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
820	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
840	00 00 00 00 00 00 00 00 00 00 00 00 00 3C 25 1F 36

The HID report can be broken down using the information in section **6 Magnetic Stripe Card Data Sent from Device to Host**, which is summarized as the **Offset** and **Usage Name** columns of **Table 9-3**. This provides a structure for organizing the raw data in the **Data** column:

Table 9-3 - Interpreting HID Data

Offset	Usage Name	Data
0	Track 1 decode status	00
1	Track 2 decode status	00
2	Track 3 decode status	00
3	Track 1 encrypted data length	3C (60 bytes, see Track 1 encrypted data below)
4	Track 2 encrypted data length	25 (37 bytes, see Track 2 encrypted data below)
5	Track 3 encrypted data length	1F (31 bytes, see Track 3 encrypted data below)
6	Card encode type (ISO/ABA)	00
7..118	Track 1 encrypted data	60 bytes, not encrypted, device is in security level 2: 25 42 35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20 20 20 20 5E 30 38 30 34 33 32 31 30 30 30 30 30 30 37 32 35 30 30 30 30 30 30 3F 00
119..230	Track 2 encrypted data	37 bytes, not encrypted, device is in security level 2: 3B 35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 3D 30 38 30 34 33 32 31 30 30 30 30 30 30 37 32 35 30 3F 00
231..342	Track 3 encrypted data	31 bytes, not encrypted, device is in security level 2: 3B 35 31 36 33 34 39 39 30 38 30 30 32 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30 30 30 3F 00
343	Card status	00 (not used, always zero)
344..347	MagnePrint status	00 00 00 00 (not available in Security Level 2)
348	MagnePrint data length	00 (Security Level 2, no MagnePrint data)

Offset	Usage Name	Data
349..476	MagnePrint data	MagnePrint not available in Security Level 2: 00
477..492	Device serial number	Not set, not filled: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
493..494	Device Encryption Status	Security Level 2, keys loaded: 00 02
495..504	DUKPT Key Serial Number (KSN) / counter	Security Level 2, not available: 00 00 00 00 00 00 00 00 00 00
505	Track 1 Masked data length	3C
506	Track 2 Masked data length	25
507	Track 3 Masked data length	1F
508..619	Track 1 Masked data	Same as encrypted data: 25 42 35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20 20 20 20 5E 30 38 30 34 33 32 31 30 30 30 30 30 30 37 32 35 30 30 30 30 30 30 3F 00
620..731	Track 2 Masked data	Same as encrypted data: 3B 35 34 35 32 33 30 30 35 35 31 32 32 37 31 38 39 3D 30 38 30 34 33 32 31 30 30 30 30 30 30 37 32 35 30 3F 00
732 to 843	Track 3 Masked data	Same as encrypted data: 3B 35 31 36 33 34 39 39 30 38 30 30 32 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30 30 30 3F 00
844..851	Encrypted Session ID	Host software didn't set, so all zeroes: 00 00 00 00 00 00 00 00

Offset	Usage Name	Data
852	Track 1 Absolute data length	3C (same as above)
853	Track 2 Absolute data length	25 (same as above)
854	Track 3 Absolute data length	1F (same as above)
855	MagnePrint Absolute data length	36 (same as above)

B.1.2 Example: Swipe Decryption, HID Device In Security Level 3 or 4 (HID Only, MSR Only)

This example shows the data received in a HID report [see section 2.1 How to Use USB Connections (USB Only)] for a device set to **Security Level 3**, KSN Count = 8. It includes steps showing how to decrypt the received data.

The raw incoming HID report is:

Byte	Content
0	00 00 00 40 28 20 00 C2 5C 1D 11 97 D3 1C AA 87 28 5D 59 A8
20	92 04 74 26 D9 18 2E C1 13 53 C0 51 AD D6 D0 F0 72 A6 CB 34
40	36 56 0B 30 71 FC 1F D1 1D 9F 7E 74 88 67 42 D9 BE E0 CF D1
60	EA 10 64 C2 13 BB 55 27 8B 2F 12 00 00 00 00 00 00 00 00 00
80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
120	4C 5D B7 D6 F9 01 C7 F0 FE AE 79 08 80 10 93 B3 DB FE 51 CC
140	F6 D4 83 E7 89 D7 D2 C0 07 D5 39 49 9B AA DC C8 D1 6C A2 00
160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
220	00 00 00 00 00 00 00 00 00 00 00 00 00 76 BB 01 3C 0D FD 81 95 F1
240	6F 2F BC 50 A3 51 71 AA 37 01 31 F8 74 42 31 3E E3 64 57 B8
260	7C 87 F9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
340	00 00 00 00 A1 05 00 00 38 47 03 57 6B C5 C2 CB 20 BC 04 C6
360	8B 5C E1 97 2A E8 9E 08 7B 1C 4D 47 D5 D0 E3 17 06 10 69 03
380	E6 0B 82 03 07 92 69 0A 57 1D B0 2D 0A 88 85 5A 35 AB B5 54
400	97 98 00 6B 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00
420	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
440	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
460	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 FF FF 98 76 54
500	32 10 E0 00 08 3C 25 1F 25 42 35 34 35 32 30 30 30 30 30 30
520	30 30 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20
540	20 20 20 5E 30 38 30 34 30 30 30 30 30 30 30 30 30 30 30 30
560	30 30 30 30 30 30 30 30 3F 00 00 00 00 00 00 00 00 00 00 00
580	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
600	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
620	3B 35 34 35 32 30 30 30 30 30 30 30 30 30 37 31 38 39 3D 30 38
640	30 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 3F 00 00
660	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
680	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
700	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
720	00 00 00 00 00 00 00 00 00 00 00 00 00 00 3B 35 31 36 33 30 30 30
740	30 35 30 30 30 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30 30
760	30 30 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
780	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
800	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
820	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
840	00 00 00 00 21 68 5F 15 8B 5C 6B E0 3C 25 1F 36

The HID report can be broken down using the information in section **6 Magnetic Stripe Card Data Sent from Device to Host**, which is summarized as the **Offset** and **Usage Name** columns of **Table 9-4**. This provides a structure for organizing the raw data in the **Data** column:

Table 9-4 - Interpreting HID Data

Offset	Usage Name	Data
0	Track 1 decode status	00
1	Track 2 decode status	00
2	Track 3 decode status	00
3	Track 1 encrypted data length	40 (64 bytes, always in multiples of 8)
4	Track 2 encrypted data length	28 (40 bytes, always in multiples of 8)
5	Track 3 encrypted data length	20 (32 bytes, always in multiples of 8)
6	Card encode type (ISO/ABA)	00
7 to 118	Track 1 encrypted data	C2 5C 1D 11 97 D3 1C AA 87 28 5D 59 A8 92 04 74 26 D9 18 2E C1 13 53 C0 51 AD D6 D0 F0 72 A6 CB 34 36 56 0B 30 71 FC 1F D1 1D 9F 7E 74 88 67 42 D9 BE E0 CF D1 EA 10 64 C2 13 BB 55 27 8B 2F 12 00
119 to 230	Track 2 encrypted data	72 4C 5D B7 D6 F9 01 C7 F0 FE AE 79 08 80 10 93 B3 DB FE 51 CC F6 D4 83 E7 89 D7 D2 C0 07 D5 39 49 9B AA DC C8 D1 6C A2 00
231 to 342	Track 3 encrypted data	76 BB 01 3C 0D FD 81 95 F1 6F 2F BC 50 A3 51 71 AA 37 01 31 F8 74 42 31 3E E3 64 57 B8 7C 87 F9 00
343	Card status	00 (not used, always zero)
344 to 347	MagnePrint status	A1 05 00 00
348	MagnePrint data length	38

Offset	Usage Name	Data
349 to 476	MagnePrint data	47 03 57 6B C5 C2 CB 20 BC 04 C6 8B 5C E1 97 2A E8 9E 08 7B 1C 4D 47 D5 D0 E3 17 06 10 69 03 E6 0B 82 03 07 92 69 0A 57 1D B0 2D 0A 88 85 5A 35 AB B5 54 97 98 00 6B 42 00
477 to 492	Device serial number	(Not set, not filled) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
493 to 494	Device Encryption Status	(Keys loaded, encrypting) 00 06
495 to 504	DUKPT Key Serial Number (KSN) / counter	FF FF 98 76 54 32 10 E0 00 08
505	Track 1 Masked data length	3C
506	Track 2 Masked data length	25
507	Track 3 Masked data length	1F
508 to 619	Track 1 Masked data	25 42 35 34 35 32 30 30 30 30 30 30 30 30 37 31 38 39 5E 48 4F 47 41 4E 2F 50 41 55 4C 20 20 20 20 20 20 5E 30 38 30 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 3F 00
620 to 731	Track 2 Masked data	3B 35 34 35 32 30 30 30 30 30 30 30 30 37 31 38 39 3D 30 38 30 34 30 30 30 30 30 30 30 30 30 30 30 30 30 3F 00
732 to 843	Track 3 Masked data	3B 35 31 36 33 30 30 30 30 35 30 30 30 34 34 35 3D 30 30 30 30 30 30 30 30 30 30 30 30 3F 00
844 to 851	Encrypted Session ID	(Host software didn't set, so decrypts to all zeroes) 21 68 5F 15 8B 5C 6B E0
852	Track 1 Absolute data length	3C
853	Track 2 Absolute data length	25

Offset	Usage Name	Data
854	Track 3 Absolute data length	1F
855	MagnePrint Absolute data length	36

To decrypt this data, the host software would first examine the KSN field FFFF9876543210E00008, and break it down into base key FFFF9876543210E and the key counter is 0x00008 (see section 6.15 **DUKPT Key Serial Number** for details). The host would use this information to calculate encryption key 27F66D5244FF621E AA6F6120EDEB427F, which is also provided in the ANSI standard documentation's examples for convenience.

There are five encrypted values: Track 1 encrypted data, track 2 encrypted data, track 3 encrypted data, encrypted MagnePrint data, and encrypted session ID. The remainder of this section details the procedure for decrypting these data values.

The track 1 encrypted data is:

C2 5C 1D 11 97 D3 1C AA
87 28 5D 59 A8 92 04 74
26 D9 18 2E C1 13 53 C0
51 AD D6 D0 F0 72 A6 CB
34 36 56 0B 30 71 FC 1F
D1 1D 9F 7E 74 88 67 42
D9 BE E0 CF D1 EA 10 64
C2 13 BB 55 27 8B 2F 12
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00

Because the **Track 1 Encrypted Data Length (HID | GATT | SLIP)** value in the incoming data says Track 1 Encrypted data is 64 bytes long, the host software can truncate the trailing blocks:

Block #	Content
1	C25C1D1197D31CAA
2	87285D59A8920474
3	26D9182EC11353C0
4	51ADD6D0F072A6CB
5	3436560B3071FC1F
6	D11D9F7E74886742
7	D9BEE0CFD1EA1064
8	C213BB55278B2F12

Section 5 **Encryption, Decryption, and Key Management** tells us to decrypt the last block first: C213BB55278B2F12 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets

E98ED0F0D1EA1064, XOR D9BEE0CFD1EA1064 gets 3030303F00000000, which is the decrypted last block.

Continuing in reverse block order, D9BEE0CFD1EA1064 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets E12DA84C41B85772, XOR D11D9F7E74886742 gets 3030373235303030, which is decrypted block 7.

Continuing in reverse block order, D11D9F7E74886742 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 0704673B0041CC2F, XOR 3436560B3071FC1F gets 3332313030303030, which is decrypted block 6.

Continuing in reverse block order, 3436560B3071FC1F TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 718DF68EC04A96FF, XOR 51ADD6D0F072A6CB gets 2020205E30383034, which is decrypted block 5.

Continuing in reverse block order, 51ADD6D0F072A6CB TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 0989597B8D3373E0, XOR 26D9182EC11353C0 gets 2F5041554C202020, which is decrypted block 4.

Continuing in reverse block order, 26D9182EC11353C0 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets BF110311E7D5453A, XOR 87285D59A8920474 gets 38395E484F47414E, which is decrypted block 3.

Continuing in reverse block order, 87285D59A8920474 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets F2692820A5E12B9B, XOR C25C1D1197D31CAA gets 3035353132323731, which is decrypted block 2.

Continuing in reverse block order, C25C1D1197D31CAA TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 2542353435323330, which is decrypted block 1.

Ordering the decrypted blocks first to last yields the following. The ASCII translation on the right shows the host ignoring the final four bytes from the HEX block because the **Track 1 Absolute Data Length (HID | GATT | SLIP)** value in the data indicates Track 1 only contains 60 characters:

HEX	ASCII
2542353435323330	%B545230
3035353132323731	05512271
38395E484F47414E	89^HOGAN
2F5041554C202020	/PAUL
2020205E30383034	^0804
3332313030303030	32100000
3030373235303030	00725000
3030303F00000000	000?

The resulting ASCII string for track 1 is:

```
%B5452300551227189^HOGAN/PAUL      ^08043210000000725000000?
```

The track 2 encrypted data is:

```
72 4C 5D B7 D6 F9 01 C7
F0 FE AE 79 08 80 10 93
B3 DB FE 51 CC F6 D4 83
E7 89 D7 D2 C0 07 D5 39
49 9B AA DC C8 D1 6C A2
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Because the **Track 2 Encrypted Data Length (HID | GATT | SLIP)** value in the incoming data says Track 2 encrypted data is 40 bytes long, the host software can truncate the trailing blocks:

Block #	Data
1	724C5DB7D6F901C7
2	F0FEAE7908801093
3	B3DBFE51CCF6D483
4	E789D7D2C007D539
5	499BAADCC8D16CA2

Section 5 Encryption, Decryption, and Key Management tells us to decrypt the last block first: 499BAADCC8D16CA2 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets D0BBE2E2FF07D539, XOR E789D7D2C007D539 gets 373235303F000000, which is the decrypted final block.

Continuing in reverse block order, E789D7D2C007D539 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 82EBCE61FCC6E4B3, XOR B3DBFE51CCF6D483 gets 3130303030303030, which is decrypted block 4.

Continuing in reverse block order, B3DBFE51CCF6D483 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets C9C39E4138B423A1, XOR F0FEAE7908801093 gets 393D303830343332, which is decrypted block 3.

Continuing in reverse block order, F0FEAE7908801093 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 47796C85E4CE30FF, XOR 724C5DB7D6F901C7 gets 3535313232373138, which is decrypted block 2.

Continuing in reverse block order, 724C5DB7D6F901C7 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 3B35343532333030, which is decrypted block 1.

Ordering the decrypted blocks first to last gives:

Appendix B - Examples

HEX	ASCII
3B35343532333030	;5452300
3535313232373138	55122718
393D303830343332	9=080432
3130303030303030	10000000
373235303F000000	7250?

The host software can ignore the last three bytes because the **Track 2 Absolute Data Length (HID | GATT | SLIP)** value in the incoming data specifies that data is 37 characters long.

The resulting ASCII string for track 2 is:

```
;5452300551227189=080432100000007250?
```

The track 3 encrypted data is:

```
76 BB 01 3C 0D FD 81 95
F1 6F 2F BC 50 A3 51 71
AA 37 01 31 F8 74 42 31
3E E3 64 57 B8 7C 87 F9
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Following the same procedures described above for track 1 and track 2 yields this ASCII string for track 3:

```
;5163499080020445=000000000000?
```

The MagnePrint encrypted data is:

```
47 03 57 6B C5 C2 CB 20
BC 04 C6 8B 5C E1 97 2A
E8 9E 08 7B 1C 4D 47 D5
D0 E3 17 06 10 69 03 E6
0B 82 03 07 92 69 0A 57
1D B0 2D 0A 88 85 5A 35
AB B5 54 97 98 00 6B 42
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```



```
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Because the **MagnePrint Data Length (HID | GATT | SLIP)** value in the incoming data says MagnePrint encrypted data is 56 bytes long, the host software can truncate the trailing blocks:

Block #	Data
1	4703576BC5C2CB20
2	BC04C68B5CE1972A
3	E89E087B1C4D47D5
4	D0E31706106903E6
5	0B82030792690A57
6	1DB02D0A88855A35
7	ABB5549798006B42

Section 5 Encryption, Decryption, and Key Management tells us to decrypt the last block first: ABB5549798006B42 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets D3B7EDDFD3045A35, **XOR** 1DB02D0A88855A35 gets CE07C0D55B810000, which is the decrypted final block.

Continuing in reverse block order, 1DB02D0A88855A35 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets B52307C37D314482, **XOR** 0B82030792690A57 gets BEA104C4EF584ED5, which is decrypted block 6.

Continuing in reverse block order, 0B82030792690A57 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets AF4EABEE4973E402, **XOR** D0E31706106903E6 gets 7FADBCE8591AE7E4, which is decrypted block 5.

Continuing in reverse block order, D0E31706106903E6 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets 269870C3659D905E, **XOR** E89E087B1C4D47D5 gets CE0678B879D0D78B, which is decrypted block 4.

Continuing in reverse block order, E89E087B1C4D47D5 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets 7B8F912DAF1B3149, **XOR** BC04C68B5CE1972A gets C78B57A6F3FAA663, which is decrypted block 3.

Continuing in reverse block order, BC04C68B5CE1972A **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets 078FD0419993F7B0, **XOR** 4703576BC5C2CB20 gets 408C872A5C513C90, which is decrypted block 2.

Continuing in reverse block order, 4703576BC5C2CB20 **TDES Decrypt** with 27F66D5244FF621E AA6F6120EDEB427F gets 01000184EA10B939, which is decrypted block 1.

Ordering the decrypted blocks first to last yields:

```
HEX
01000184EA10B939
408C872A5C513C90
C78B57A6F3FAA663
CE0678B879D0D78B
7FADBCE8591AE7E4
BEA104C4EF584ED5
CE07C0D55B810000
```

The host software can ignore the last three bytes because the **MagnePrint Absolute Data Length (HID | TLV | GATT | SLIP)** value in the incoming data specifies that data is 54 characters long.

The resulting decrypted MagnePrint data is:

```
01000184EA10B939408C872A5C513C90C78B57A6F3FAA663CE0678B879D0D78B7FADBC
E8591AE7E4BEA104C4EF584ED5CE07C0D55B81
```

The Encrypted Session ID data is:

```
21 68 5F 15 8B 5C 6B E0
```

This is a simple eight byte block, so the host software can simply decrypt it with the appropriate key. 21685F158B5C6BE0 TDES Decrypt with 27F66D5244FF621E AA6F6120EDEB427F gets 0000000000000000. It contains all zeroes because the host software did not specify a session ID.

B.1.3 Example: Configuring a Device Before Encryption Is Enabled (HID Only)

This example configures the device to use the USB-HID data format (see section 2.1.3 How to Receive Data On the USB Connection).

```
; This script demonstrates configuration commands for HID mode.
; It assumes the device is at Security Level 2 and that the Device
; Serial Number has never been set.
00 01 10      ; Get current interface
Request       : CMND=00, LEN=01, DATA=10
Response      : RC= 00, LEN=01, DATA=01

01 02 10 00   ; Set Interface to HID
Request       : CMND=01, LEN=02, DATA=10 00
Response      : RC= 00, LEN=00, DATA=

02 00         ; Reset so changes take effect
Request       : CMND=02, LEN=00, DATA=
Response      : RC= 00, LEN=00, DATA=

Delay         : (waited 5 seconds)

00 01 10      ; Get current interface (should return 0)
Request       : CMND=00, LEN=01, DATA=10
Response      : RC= 00, LEN=01, DATA=00
```

B.1.4 Example: Changing from Security Level 2 to Security Level 3

```

; This script demonstrates changing from Security Level 2 to Security
Level 3.
; It assumes the device is at Security Level 2 with the ANS X9.24
Example
; key loaded and the KSN counter set to 1.
09 00          ; Get current KSN (should be FFFF9876543210E00001)
Request        : CMND=09, LEN=00, DATA=
Response       : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 01

; For KSN 1, MAC Key: 042666B4918430A3 68DE9628D03984C9
;
; The command to change Security Level looks like: 15 05 03 nnnnnnnn
; where nnnnnnnn is the MAC.
;
; The data to be MACd is: 15 05 03
; Data to be MACd must be in blocks of eight bytes, so we left justify
and
; zero fill the block to get: 15 05 03 00 00 00 00 00 (This is the
block to MAC)
; For convenience show it as the compacted form: 1505030000000000
;
; The MAC algorithm run with this data uses the following
cryptographic
; operations:
;
; Single DES Encrypt the data to be MACd with the left half of the
MAC Key:
; 1505030000000000 1DES Enc with 042666B4918430A3 =
BFBA7AE4C1597E3D
;
; Single DES Decrypt the result with the right half of the MAC Key:
; BFBA7AE4C1597E3D 1DES Dec with 68DE9628D03984C9 =
DA91AB9A8AD9AB4C
;
; Single DES Encrypt the result with the left half of the MAC Key:
; DA91AB9A8AD9AB4C 1DES Enc with 042666B4918430A3 =
E7E2FA3882BB386C
;
; The leftmost four bytes of the final result are the MAC = E7E2FA38
;
; Send the MACd Set Security Level command
15 05 03 E7E2FA38
Request        : CMND=15, LEN=05, DATA=03 E7 E2 FA 38
Response       : RC= 00, LEN=00, DATA=

02 00          ; Reset so changes take effect
Request        : CMND=02, LEN=00, DATA=
Response       : RC= 00, LEN=00, DATA=

Delay          : (waited 5 seconds)

```

Appendix B - Examples

```
09 00      ; Get current KSN (should be FFFF9876543210E00002)
Request    : CMND=09, LEN=00, DATA=
Response   : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 02

15 00      ; Get current Security Level (Should be 03)
Request    : CMND=15, LEN=00, DATA=
Response   : RC= 00, LEN=01, DATA=03
```

B.1.5 Example: Changing from Security Level 2 to Security Level 4 (MSR Only)

```

; This script demonstrates changing from Security Level 2 to Security
Level 4.
; It assumes the device is at Security Level 2 with the ANS X9.24
Example
; key loaded and the KSN counter set to 1.
09 00          ; Get current KSN (should be FFFF9876543210E00001)
Request        : CMND=09, LEN=00, DATA=
Response       : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 01

; For KSN 1, MAC Key: 042666B4918430A3 68DE9628D03984C9
;
; The command to change Security Level looks like: 15 05 04 nnnnnnnn
; where nnnnnnnn is the MAC.
;
; The data to be MACd is: 15 05 04
; Data to be MACd must be in blocks of eight bytes, so we left justify
and
; zero fill the block to get: 15 05 04 00 00 00 00 00 (This is the
block to MAC)
; For convenience show it as the compacted form: 1505040000000000
;
; The MAC algorithm run with this data uses the following
cryptographic
; operations:
;
; Single DES Encrypt the data to be MACd with the left half of the
MAC Key:
; 1505040000000000 1DES Enc with 042666B4918430A3 =
644E76C88FFA0044
;
; Single DES Decrypt the result with the right half of the MAC Key:
; 644E76C88FFA0044 1DES Dec with 68DE9628D03984C9 =
DEAC363779906C06
;
; Single DES Encrypt the result with the left half of the MAC Key:
; DEAC363779906C06 1DES Enc with 042666B4918430A3 =
2F38A60E3F6AD6AD
;
; The leftmost four bytes of the final result are the MAC = 2F38A60E
;
; Send the MACd Set Security Level command
15 05 04 2F38A60E
Request        : CMND=15, LEN=05, DATA=04 2F 38 A6 0E
Response       : RC= 00, LEN=00, DATA=

02 00          ; Reset so changes take effect
Request        : CMND=02, LEN=00, DATA=
Response       : RC= 00, LEN=00, DATA=

Delay          : (waited 5 seconds)

```

Appendix B - Examples

```
09 00      ; Get current KSN (should be FFFF9876543210E00002)
Request    : CMND=09, LEN=00, DATA=
Response   : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 02

15 00      ; Get current Security Level (Should be 04)
Request    : CMND=15, LEN=00, DATA=
Response   : RC= 00, LEN=01, DATA=04
```

B.1.6 Example: Changing from Security Level 3 to Security Level 4 (MSR Only)

```

; This script demonstrates changing from Security Level 3 to Security
Level 4.
; It assumes the device is at Security Level 3 with the ANS X9.24
Example
; key loaded and the KSN counter set to 2.
09 00          ; Get current KSN (should be FFFF9876543210E00002)
Request       : CMND=09, LEN=00, DATA=
Response      : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 02

; For KSN 2, MAC Key: C46551CEF9FDDBB0 AA9AD834130DC4C7
;
; The command to change Security Level looks like: 15 05 04 nnnnnnnn
; where nnnnnnnn is the MAC.
;
; The data to be MACd is: 15 05 04
; Data to be MACd must be in blocks of eight bytes, so we left justify
and
; zero fill the block to get: 15 05 04 00 00 00 00 00 (This is the
block to MAC)
; For convenience show it as the compacted form: 1505040000000000
;
; The MAC algorithm run with this data uses the following
cryptographic
; operations:
;
; Single DES Encrypt the data to be MACd with the left half of the
MAC Key:
; 1505040000000000 1DES Enc with C46551CEF9FDDBB0 =
735323A914B9482E
;
; Single DES Decrypt the result with the right half of the MAC Key:
; 735323A914B9482E 1DES Dec with AA9AD834130DC4C7 =
390E2E2AC8CB4EE6
;
; Single DES Encrypt the result with the left half of the MAC Key:
; 390E2E2AC8CB4EE6 1DES Enc with C46551CEF9FDDBB0 =
D9B7F3D8064C4B26
;
; The leftmost four bytes of the final result are the MAC = D9B7F3D8
;
; Send the MACd Set Security Level command
15 05 04 D9B7F3D8
Request       : CMND=15, LEN=05, DATA=04 D9 B7 F3 D8
Response      : RC= 00, LEN=00, DATA=

02 00          ; Reset so changes take effect
Request       : CMND=02, LEN=00, DATA=
Response      : RC= 00, LEN=00, DATA=

Delay        : (waited 5 seconds)

```


Appendix B - Examples

```
09 00      ; Get current KSN (should be FFFF9876543210E00003)
Request    : CMND=09, LEN=00, DATA=
Response   : RC= 00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 03

15 00      ; Get current Security Level (Should be 04)
Request    : CMND=15, LEN=00, DATA=
Response   : RC= 00, LEN=01, DATA=04
```

B.1.7 Example: Authentication (MSR Only)

In this example, the device is already in **Security Level 3** or **Security Level 4**. The script puts the device into Authenticated Mode, leaves it in that mode for a time, then deactivates it.

```

; This example demonstrates the Authentication Sequence.
; It is not scripted, some of the data is deliberately randomized.
This
; makes it impossible for a simple script to produce the correct
results.
; As an example it shows all the steps in authentication and
deactivation.

; It assumes the device is at Security Level 4, with the DUKPT KSN
; counter set to 2.

09 00          ; Get current KSN (should be FFFF9876543210E00002)

; Send the Activate Authenticated Mode command (4 minutes)
10 02 00F0
Request       : CMND=10, LEN=02, DATA=00 F0
Response      : RC= 00, LEN=1A, DATA=FF FF 98 76 54 32 10 E0 00 03 AA
AA AA AA AA AA AA DD DD DD DD DD DD DD DD
                                |----- Current KSN -----| |--
-- Challenge 1 ----| |---- Challenge 2 ----|
Response      : RC= 00, LEN=1A, DATA=FF FF 98 76 54 32 10 E0 00 03 BE
5C 98 35 17 7E 45 2A A7 2D 2D B2 36 BF 29 D2
; Challenge 1 Encrypted: BE5C9835177E452A
; Challenge 2 Encrypted: A72D2DB236BF29D2

; Note that the KSN now ends with a counter of 3!
; Decrypt Challenge 1 using variant of Current Encryption Key
; (Current Encryption Key XOR with F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0)
;
; Current Key   0DF3D9422ACA561A 47676D07AD6BAD05
; XOR          F0F0F0F0F0F0F0F0 F0F0F0F0F0F0F0F0
; =            FD0329B2DA3AA6EA B7979DF75D9B5DF5
;
; BE5C9835177E452A TDES Decrypt with FD0329B2DA3AA6EA
B7979DF75D9B5DF5 = 7549AB6EB4840003
;
; Note that the final two bytes of the result = 0003, matching the
KSN as
; transmitted in the clear. This provides Authentication to the
host that
; the device is what it claims to be (proves key knowledge).
;
; Decrypt Challenge 2 using Current Encryption Key variant as above
; A72D2DB236BF29D2 TDES Decrypt with FD0329B2DA3AA6EA
B7979DF75D9B5DF5 = 34DB9230698281B4
;
;

```

```

; Build an Activation Challenge Reply command (cmd, len, cryptogram)
; 11 08 XXXXXXXXXXXXXXXXXXXX
;
; The clear text input for the cryptogram is composed of the first
six bytes
; of the decrypted Challenge 1 followed by two bytes specifying how
long to
; stay in the Authenticated Mode.
;
; CCCCCCCCCCCC TTTT
;
; Time examples:
; For 30 seconds use 001E
; For 99 seconds use 0063
; For 480 seconds use 01E0
; For 1200 seconds use 04B0
;
; These values are concatenated to form an eight byte block, we will
use 480
; seconds:
;
; CCCCCCCCCCCC01E0
;
; The block is encrypted using a variant of the Current Encryption
Key
; (Current Encryption Key XOR with 3C3C3C3C3C3C3C3C3C3C3C3C3C3C3C3C)
;
; Current Key 0DF3D9422ACA561A 47676D07AD6BAD05
; XOR 3C3C3C3C3C3C3C3C 3C3C3C3C3C3C3C3C
; = 31CFE57E16F66A26 7B5B513B91579139
;
; 7549AB6EB48401E0 TDES Enc with 31CFE57E16F66A26 7B5B513B91579139
= A30DDE3BFD629ACD
;
; Send the Activation Challenge Reply Command
11 08 A30DDE3BFD629ACD
;
; Build a Deactivate Authenticated Mode command (cmd, len, cryptogram)
; 12 08 XXXXXXXXXXXXXXXXXXXX
;
; The clear text input for the cryptogram is composed of the first
seven bytes
; of the decrypted Challenge 2 followed by one byte specifying
whether to
; increment the DUKPT KSN or not (00 = no increment, 01 = increment).
;
; DDDDDDDDDDDDDD II
;
; These values are concatenated to form an eight byte block, we will
specify
; No Increment:

```

```
;
;   DDDDDDDDDDDDDDD00
;
;   The block is encrypted using a variant of the Current Encryption
Key
;   (Current Encryption Key XOR with 3C3C3C3C3C3C3C3C3C3C3C3C3C3C3C3C)
;
;   Current Key   0DF3D9422ACA561A 47676D07AD6BAD05
;               XOR   3C3C3C3C3C3C3C3C 3C3C3C3C3C3C3C3C
;               =   31CFE57E16F66A26 7B5B513B91579139
;
;   34DB923069828100 TDES Enc with 31CFE57E16F66A26 7B5B513B91579139
= CA CB BD 5F 58 D5 C9 50
;
;   Send the Deactivate Authenticated Mode command
12 08 CACBBD5F58D5C950
```

B.2 About the SDKs and Additional Examples

MagTek provides SDKs and corresponding documentation for many programming languages and operating systems that enable software developers to quickly develop custom host software that communicates with this device, without having to deal with the complexities of platform APIs for direct communication across the various available connection types, connecting using the various available communication protocols, and parsing the various available data formats.

The SDKs and corresponding documentation include:

- Functions for sending the direct commands described in this manual
- Wrappers for commonly used commands and properties that further simplify development
- Detailed compilable examples of processing incoming payment data and using the direct commands and properties described in this manual

To download the SDKs and documentation, search www.magtek.com for “SDK” and select the SDK and documentation for the programming languages and platforms you need, or contact MagTek Support Services for assistance.

Appendix C Identifying ISO/ABA and AAMVA Cards For Masking (MSR Only)

C.1 ISO/ABA Financial Card

The device uses the rules below to determine if a card is an ISO/ABA card (per *ISO 7811-2,2001*), which affects incoming **Card Encode Type (HID | TLV | GATT | SLIP)** as well as the masking used for **Masked Track Data**. ISO defines a particular and different bit-level character encoding format of the data on each of the three tracks of the card. The format of the card depends on decisions made by the entity that issued the card. For example, some organizations may choose to use the ISO Track 1 encoding format for Track 2 data, or other permutations that do not conform to the standard. The device only considers ISO Financial masking for cards it classifies as ISO, which it determines according to the following rules:

- 1) If the low level decoding algorithm determines the bit level character encoding for every track conforms to the ISO format defined for that track, the card is classified as ISO. Otherwise the device attempts to classify the card as an **AAMVA Driver's License**, and if the card fails that test, the device classifies the card as **Other**. A properly encoded ISO card has the following properties:
 - a) At least one track must be decodable.
 - b) Track 1 must be 7 bits per character.
 - c) If Track 2 or Track 3 exist, they must be 5 bits per character.
- 2) If the device determines the card is ISO encoded, it then determines the masking behavior for each track independently. One track may qualify for masking and another may not, according to the following rules.
- 3) For Track 1, the device's intent is to send the card's Format Code unmasked, the PAN partially masked, the Name and Expiration Date unmasked, and the rest of the track masked, with exceptions:
 - a) The Service Code is always unmasked on newer devices.
 - b) If the card's Format Code, PAN, Name, or Expiration Date are not correctly structured, the device transmits the rest of the track unmasked starting with the point of discrepancy. The device defines "correct structure" for Track 1 as follows:
 - i) The card's Format Code, PAN, Name, or Expiration Date do not contain the '?' character (End Sentinel).
 - ii) The Format Code is the first character on the track and is the character 'B'.
 - iii) The PAN has a maximum of 19 digits and ends with character '^' (Field Separator).
 - iv) The Cardholder Name has a maximum of 26 characters and is ended by the character '^' (Field Separator).
 - v) The Expiration Date has 4 characters.
 - vi) The Service Code has 3 characters.
- 4) For Track 2, the device's intent is to send the PAN partially masked, the Expiration Date unmasked, and most of the rest of the track masked, with exceptions:
 - a) The Service Code is always unmasked on newer devices.
 - b) If the PAN or Expiration Date are not correctly structured, the device sends the rest of the track unmasked starting at the point of discrepancy. The device defines "correct structure" as follows:
 - i) The PAN or Expiration Date does not contain the '?' character (End Sentinel).
 - ii) The PAN has a maximum of 19 digits and ends with the character '=' (Field Separator).
 - iii) The Expiration Date has 4 characters.
 - iv) The Service Code has 3 characters.

- 5) For Track 3, the device's intent is to send the PAN partially masked and the rest of the track masked, with exceptions:
 - a) If the PAN is not correctly structured, the device sends the rest of the track unmasked, starting at the point of discrepancy. The device defines "correct structure" as follows:
 - i) The PAN does not contain the '?' character (End Sentinel).
 - ii) The PAN has a maximum of 19 digits and ends with character '=' (Field Separator).

C.2 AAMVA Driver's License

The device uses the following rules to determine if a card is an AAMVA card:

- 1) If the device reads three tracks of data and Track 1 is formatted per ISO Track 1 rules, Track 2 is formatted per ISO Track 2 rules, and Track 3 is formatted per ***ISO Track 1*** [sic.] rules, the card is considered to be an AAMVA card. Some MagTek devices do not support reading of Track 3, so this rule does not apply on such devices.
- 2) If a low level decoding algorithm finds data for the available tracks to be in the ISO format particular to each track, and Track 2 contains a correctly structured PAN field whose first 6 digits are "604425" or contain values in the range "636000" to "636062" inclusive, the card is considered to be an AAMVA card.

AAMVA card masking, when enabled, works as follows:

- 1) The device sends track 1 and track 3 entirely masked; all character positions are filled with zeroes.
- 2) Track 2 is treated as follows:
 - a) The device's intent is to send the Driver License ID (DLID) partially masked, the Expiration Date unmasked, the Birth Date unmasked, and the rest of the track masked.
 - b) If the DLID, Expiration Date, or Birth Date are not correctly structured, the rest of the track, starting at the point of discrepancy, is sent unmasked. The device defines "correctly structured" as follows:
 - i) If the DLID, Expiration Date, or Birth Date contain the '?' character (End Sentinel), the field is not correctly structured.
 - ii) A correctly structured DLID has a maximum of 19 digits and is terminated by the character '=' (Field Separator).
 - iii) A correctly structured Expiration Date has 4 characters.
 - iv) A correctly structured Birth Date has 8 characters.

Appendix D EMV Message Formats (EMV Only)

D.1 ARQC Messages (EMV Only)

This section gives the format of the ARQC Message delivered in **Notification 0x0303 - ARQC Message**. The contents of the ARQC Message is slightly different depending on whether the device is set to **Security Level 2** (not encrypting) or **Security Level 3** (encrypting). Support for EMV transactions at **Security Level 2** is only available on mDynamo.

D.1.1 ARQC Message Format Security Level 2

When the device is set to **Security Level 2** (not encrypting), the ARQC Message TLV data object contains the following:

```
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> /* Masked T2 PICC/ICC Data */
            DFDF52<len><value> /* card type */
            <tags defined by DFDF02 >

(Buffer if any to be a multiple of 8 bytes)

CBC-MAC (4 bytes reserved, not calculated)
```

The device populates TLV data object DFDF53 with one of the following fallback indicators:

- 0x00 = No fallback or missing tag

The device populates TLV data object DFDF52 with one of the following card types:

- 0x00 = Other
- 0x01 = Financial
- 0x02 = AAMVA
- 0x03 = Manual
- 0x04 = Unknown
- 0x05 = ICC
- 0x06 = Contactless ICC - EMV
- 0x07 = Financial MSR and ICC
- 0x08 = Contactless ICC - MSD

The device constructs the contents of tag DFDF4D, using EMV transaction data to emulate track 2 data as though it came from an ISO/ABA magnetic stripe card. Much of the data is masked; the device sends a specified mask character instead of the actual character from the transaction. The device provides masking settings in **Property 0x07 - ISO Track Mask**, which allows the host software to specify

masking details for the Primary Account Number, the masking character to be used, and whether a correction should be applied to make the Mod 10 (Luhn algorithm) digit at the end of the PAN be correct.

Table 9-5 provides an example of track 2 data as it would appear if the device sent it in the clear. **Table 9-6** shows the same data as it might appear with a specific set of masking rules applied.

Table 9-5 – Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted

Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted	
Track 2	;6011000995500000=15121015432112345678?

Table 9-6 – Sample ISO/ABA Swiped Track Data, Masked

Sample ISO/ABA Swiped Track Data, Masked	
Track 2	;6011000020000000=15120000000000000000?

Table 9-7 shows an example of track 2 data using unmasked placeholders to make it easier to see the relative positions of the values embedded in the track data, and can be interpreted as follows:

- [?] and [] are Sentinels / delimiters.
- The string of [5]s is the Account Number / PAN.
- The string of [3]s is the Expiration Date.
- The string of [8]s is the Service Code.
- The remaining characters ([0]s, [4]s, and [6]) are Discretionary Data, which is of varying length and content and comes from the card, and must be interpreted according to the standards established by issuers, payment brands, and so on.

Table 9-7 – Example Generic ISO/ABA Track Data Format

Generic ISO/ABA Track Data Format	
Track 2 Data	;5555555555555555=33338880004444006?

The device masks the data as follows:

- The number of initial characters and trailing characters specified by **Property 0x07 - ISO Track Mask** is sent unmasked. If Mod 10 correction is specified (see **Property 0x07 - ISO Track Mask**), all but one of the intermediate characters of the PAN are set to zero; one of them is set such that last digit of the PAN calculates an accurate Mod 10 check of the rest of the PAN as transmitted. If the Mod 10 correction is not specified, all of the intermediate characters of the PAN are set to the specified mask character.
- The Expiration Date is transmitted unmasked.
- The Service Code is always unmasked on newer devices.
- All Field Separators are sent unmasked.
- All other characters are set to the specified mask character.

D.1.2 ARQC Message Format Security Level 3

When the device is set to **Security Level 3** (encrypting), the ARQC Message TLV data object contains the following:

```
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> /* Masked T2 PICC/ICC Data */
      DFDF52<len><value> /* card type */
      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>

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The values inside tags DFDF52, DFDF53, and DFDF4D are fully described in section **D.1.1**.

The device encrypts the Value inside data container DFDF59 using the **Data Encryption, request or both ways** variant [or other variant depending on **Property 0x67 - EMV Data Encryption Variant (EMV Only)**] of the current DUKPT 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. After the host decrypts it, DFDF59 contains a list of TLV data objects defined by terminal setting DFDF02 or DFDF08 is card type is contactless-MSD. For example:

```
FC<len> /* container for encrypted generic data *
  <tags defined by DFDF02 or DFDF08>
  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>
```

```
Data><len><val>          DFDF43 <Magneprint Status
                           DFDF50 (MSR KSN Data)<len><val> /*sent
in the clear*/           DFDF51 (MSR EncryptionType)<len><val>
                           <Padding to force DFDF59 plus padding to be a
multiple of 8 bytes>
```

D.2 ARPC Response from Online Processing (EMV Only)

This section specifies the format of the data for **Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)**. The host sends this request to the device in response to **Notification 0x0303 - ARQC Message**.

An ARPC Response is a TLV data object with the following contents:

```
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
    (ARPC padding, if any, to be a multiple of 8 bytes)
CBC-MAC (4 bytes, reserved, must be sent to the device, however, the
device does not check for the properly calculated CBC-MAC)
```

D.3 Transaction Result Messages (EMV Only)

This section specifies the format for data the device sends using **Notification 0x0304 - Transaction Result Message**. The format is controlled by **Property 0x68 – EMV Message Format**.

TLV data object DFDF1A contains one of the following Transaction Status values:

- 0x00 = Approved
- 0x01 = Declined
- 0x02 = Error
- 0x10 = Cancelled by Host
- 0x1E = Manual Selection Cancelled by Host
- 0x1F = Manual Selection Timeout
- 0x21 = Waiting for Card Cancelled by Host
- 0x22 = Waiting for Card Timeout
- 0x23 = Cancelled by Card Swipe (MSR Only)
- 0xFF = Unknown

The format of Transaction Result messages depends on whether the device is set to **Security Level 2** (not encrypting) or **Security Level 3** (encrypting). Support for EMV transactions at **Security Level 2** is only available on mDynamo.

D.3.1 Transaction Result Message Format Security Level 2

When the device is set to **Security Level 2** (not encrypting), the Transaction Result TLV data object contains the following:

```
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 */
        DFDF1A - Transaction Status
        DFDF1B - Additional Transaction Information

      F2<len> /* container for Transaction Data */
        /* Data tags (defined by DFDF17) */

      F3<len> /* container for Reversal Data, if any */
        /* Reversal Data tags (defined by DFDF05) */

      F7<len> /* container for Merchant Data */
        /* Merchant Data tags */
        5F25<len> /* Application Effective Date */
        5F24<len> /* Application Expiration Date */
        89<len> /* Authorization Code */
        5F2A<len> /* Transaction Currency Code */
        9F02<len> /* Amount, Authorized */
        9F03<len> /* Amount, Other */
        9F06<len> /* Application Identifier */
        9F12<len> /* Application Preferred Name */
        9F1C<len> /* Terminal Identification */
        9F39<len> /* POS Entry Mode */
        9C<len> /* Transaction Type */
        9F34<len> /* Cardholder Verification Results */
        5F57<len> /* Account Type */
        5F20<len> /* Cardholder Name */
        DFDF4D<len> /* Masked T2 PICC/ICC Data */

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The value inside tag DFDF4D is fully described in section **D.1.1**.

D.3.2 Transaction Result Message Format Security Level 3

When the device is set to **Security Level 3** (encrypting), the Transaction Result TLV data object contains the following:

```
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 */
        DFDF1A - Transaction Status
        DFDF1B - Additional Transaction Information

      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 */
        5F25<len> /* Application Effective Date */
        5F24<len> /* Application Expiration Date */
        89<len> /* Authorization Code */
        5F2A<len> /* Transaction Currency Code */
        9F02<len> /* Amount, Authorized */
        9F03<len> /* Amount, Other */
        9F06<len> /* Application Identifier */
        9F12<len> /* Application Preferred Name */
        9F1C<len> /* Terminal Identification */
        9F39<len> /* POS Entry Mode */
        9C<len> /* Transaction Type */
        9F34<len> /* Cardholder Verification Results */
        5F57<len> /* Account Type */
        5F20<len> /* Cardholder Name */
        DFDF4D<len> /* Masked T2 PICC/ICC Data */

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The value inside tag DFDF4D is fully described in section **D.1.1**.

The device encrypts the Value inside data container DFDF59 using the **Data Encryption, request or both ways** variant [or other variant depending on **Property 0x67 - EMV Data Encryption Variant (EMV Only)**] of the current DUKPT 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. After the host

decrypts it, DFDF59 contains a list of TLV data objects defined by terminal setting DFDF17 and DFDF05. For example:

```
FC<len>/* container for encrypted generic data */
  F2<len>/* container for Transaction Data */
    ... /* Data tags (defined by DFDF17) */
  F3<len>/* container for Reversal Data, if any */
    ... /* Reversal Data tags (defined by DFDF05)*/
```


Appendix E EMV Terminal and Application Settings (EMV Only)

E.1 EMV Common Settings

This section lists settings that are common across all EMV databases on the device.

E.1.1 EMV Common Terminal Settings and Defaults

This section lists the default EMV Terminal Settings shared across all terminal databases on the device. For information about reading and changing these settings, see section **8.4.8 Extended Command 0x0306 - Read Terminal Configuration** and section **8.4.7 Extended Command 0x0305 - Modify Terminal Configuration**.

Table 9-8 - EMV Common Terminal Settings

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
5F2A	08 40	0x02	MagTek	Transaction Currency Code. Valid values are the numerical codes from <i>ISO 4217 Codes for the representation of currencies</i> , for example: <ul style="list-style-type: none"> 0x0000 = Use Selected Application's Currency Code Terminal Setting 0x0840 = US Dollar 0x0978 = Euro
5F36	02	0x01	MagTek	Transaction Currency Exponent
9F1A	08 40	0x02	MagTek	Terminal Country Code. The device's terminal country codes are numeric and derived from <i>ISO 3166-1</i> , for example: <ul style="list-style-type: none"> 0840 = United States 0250 = France 0380 = Italy 0724 = Spain 0276 = Germany
9F1C	31 31 32 32 33 33 34 34	0x08	MagTek	Terminal Identification
9F4E	30 30 30 30 30 30 30	0x28	MagTek	Merchant Name and Location
DFDF14	00 00 75 30	0x04	MagTek	Socket Timeout for Online Processing (ms)
DFDF15	00 00 00 01	0x04	MagTek	Socket Retries (number of connection retries in Online Processing)

E.1.2 EMV Common Application Settings and Defaults

There are no default EMV Application Settings shared across all application databases on the device.

E.2 EMV Contact Settings (Contact Only)

E.2.1 EMV Contact Terminal Settings and Defaults (Contact Only)

This section lists the default EMV Contact Terminal default settings. For information about reading and changing these settings, see section **8.4.8 Extended Command 0x0306 - Read Terminal Configuration** and section **8.4.7 Extended Command 0x0305 - Modify Terminal Configuration**.

Table 9-9 - EMV Contact Terminal Settings

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Terminal Settings from section E.1.1				
9F15	30 30	0x02	MagTek	Merchant Category Code
9F16	30 30 30 30 30 30 30	0x0F	MagTek	Merchant Identifier
9F33	20 28 C8	0x03	MagTek	Terminal Capabilities (Set by Terminal Configuration, see section 8.4.17)
9F35	21	0x01	MagTek	Terminal Type (Set by Terminal Configuration, see section 8.4.17)
9F3C	09 98	0x02	MagTek	Transaction Reference Code
9F3D	02	0x01	MagTek	Transaction Currency Exponent
9F40	72 00 00 B0 01	0x05	MagTek	Additional Terminal Capabilities (Set by Terminal Configuration, see section 8.4.17)
DFDF01	A0 00 00 00 04 F8 00 10 00	0x09	MagTek	Certificate Revocation List
DFDF02	9A DF DF 28 9F 02 5A 89 9F 10 9F 15 9F 16 9F 4E 82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 26 9F 27 9F 36 9C 9F 33 9F 34 9F 37 9F 39 9F 40 95 9B 9F 5B DF DF 00 9F 1E 9F 1A 5F 2A 9F 01 9F 21 8A DF 81 20 DF 81 21 DF 81 22 5F 20 50 5F 34 84 9F 03 9F 09 9F 1E 9F 35 9F 41 9F 53 F4	0x81	MagTek	Online message for EMV transaction

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
DFDF05	9A 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 9F 21 9C	0x80	MagTek	Reversal message for EMV transaction
DFDF06	8A 91	0x02	MagTek	Tags participating in online response
DFDF16	00 00 00 80	0x04	MagTek	Maximum length of issuer script (Read Only)
DFDF17	9A DF DF 28 9F 02 9F 03 5A 89 9F 10 9F 15 9F 16 9F 4E 82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 26 9F 27 9C 9F 33 9F 34 9F 35 9F 36 9F 37 9F 39 9F 40 9F 41 9F 53 95 9B 9F 5B DF DF 00 9F 1E 9F 1A 5F 2A 9F 01 8A DF 81 20 DF 81 21 DF 81 22 5F 20 5F 34 9F 09 84	0x80	MagTek	EMV Transaction Result Message Tags

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
DFDF20	43 28	0x02	Read Only	<p>Terminal Features, read only</p> <p>Byte 1: Bit 8 TAC/IAC-Default process when unable to go online Bit 7 Manual Language Selection Enabled Bit 6 Referrals are supported Bit 5 CDA Failure detected prior to TAA is enabled Bit 4/Bit 3 0b00 = CDA Mode 1 is enabled, 0b01 = CDA Mode 2 is enabled, 0b10 = CDA Mode 3 is enabled, 0b11 = CDA Mode 4 is enabled Bit 2 Cardholder Confirmation is enabled Bit 1 EMV Language Selection is enabled</p> <p>Byte 2: Bit 8 RFU Bit 7 'Forced Acceptance' is enabled Bit 6 'Application Preferred Order' is enabled Bit 5 'Transaction log' is enabled Bit 4 'Revocation of Issuer Public Key' is enabled Bit 3 'Account Type selection' is enabled Bit 2 'Subsequent Bypass PIN Entry' is enabled Bit 1 'Bypass PIN Entry' is enabled</p>
DFDF26	4D 41 47 54 45 4B 20 44 45 46 41 55 4C 54	0x10	MagTek	EMV Database Label
DFDF5B	0C	0x01	MagTek	Terminal Capabilities for Purchase transaction
DFDF5C	02	0x01	MagTek	Terminal Capabilities for Cashback transaction
DFDF6E	0C	0x01	MagTek	Terminal Capabilities for Payment transaction
DFDF75	0C	0x01	MagTek	Terminal Capabilities for Inquiry (Not Supported)
DFDF76	0C	0x01	MagTek	Terminal Capabilities for Transfer (Not Supported)

E.2.2 EMV Contact Application Settings and Defaults (Contact Only)

This section lists the default EMV Contact Application Settings. For information about reading and changing these settings, see section **8.4.10 Extended Command 0x0308 - Read Application Configuration** and section **8.4.9 Extended Command 0x0307 - Modify Application Configuration**.

Table 9-10 - EMV Contact Application Slot 1 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 25 01	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 25 01	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-11 - EMV Contact Application Slot 2 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 06 20 06 20	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 06 20 06 20	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	00 00 00	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 AC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-12 - EMV Contact Application Slot 3 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 01 52 30 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 01 52 30 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 27 10	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 00 00 20 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC E0 9C F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-13 - EMV Contact Application Slot 4 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 98 08 40	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 98 08 40	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 00 00 20 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC E0 9C F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-14 - EMV Contact Application Slot 5 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 02 77 10 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 02 77 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 F8 A8 F0	0x05	MagTek	Terminal Action Code - Default
DF8121	10 10 58 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC F8 E4 B8 70	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-15 - EMV Contact Application Slot 6 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 65 10 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 65 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	02 00	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 60 24 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 60 AC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-16 - EMV Contact Application Slot 7 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 04 10 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 B8 A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 B8 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-17 - EMV Contact Application Slot 8 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 04 30 60	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 30 60	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 BC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-18 - EMV Contact Application Slot 9 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 04 22 03	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 22 03	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 BC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-19 - EMV Contact Application Slot 10 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 03 33 01 01 01	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 01	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-20 - EMV Contact Application Slot 11 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 03 33 01 01 02	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 02	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-21 - EMV Contact Application Slot 12 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 03 33 01 01 03	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 03	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-22 - EMV Contact Application Slot 13 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 03 10 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-23 - EMV Contact Application Slot 14 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 03 20 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 20 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-24 - EMV Contact Application Slot 15 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	A0 00 00 00 03 30 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 30 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 9-25 - EMV Contact Application Slot 16 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All EMV Common Application Settings and Defaults from section E.1.2				
84	00	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	00	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF67	01	0x01	MagTek	MSR Fallback Supported (Not Supported)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)