MAGTEK®
SECURITY FROM THE INSIDE

oDynamo

OEM Hybrid Insertion Secure Card Reader Authenticator for Unattended Terminals

Programmer's Reference (COMMANDS)

August 2017

Document Number:
D998200162-12

REGISTERED TO ISO 9001:2008

Table 1.1 - Revisions

Rev Number	Date	Notes
10	May 10, 2017	Initial release derived from ***D100003652*** rev 10 release
11	Jul 5, 2017	Refresh from ***D100003652*** rev 11 release: Update **Table 1-1 - Device Features**; Remove Set Factory Defaults command; Update Command 0x03::0x60 - Set/Get Ethernet Configuration (Ethernet Only); Add whitelist functionality to **Command 0x03::0x80 - Get PAN / MSR Whitelist**; Update values available for **Command 0x03::0x71 - Set Device Configuration**; Remove PIN and Clear Text functions from **Application Group 0x04 - Magnetic Stripe Reader (MSR) Messages**; Add PAN and whitelist info to **Notification 0x04::0x11 - MSR Card Data Available** and **Command 0x05::0x01 - Request PAN**; Spec changed from using a connected EPP to providing PAN to an external device, which led to removing Application Group 9 EPP Commands, adding **Command 0x05::0x01 - Request PAN**; Add **Notification 0x07::0x88 - EMV L2 PIN CVM Request**; Misc. clarifications and corrections.
12	Aug 31, 2017	Refresh from ***D100003652*** rev 12 version 36 draft: Add **Command 0x01::0x53 - Activate Device**, **Command 0x08::0x04 - Read Dismount Switch State**; **Add Command 0x08::0x03 - Re-Activate Device** to customer-facing command set; Update key IDs in response of **Command 0x02::0x0B - Get Challenge**; Remove DF50 Device State from **Data Object F1 - Device Status** and update **Data Object DF51 Device Status**; Deprecate Application Group 0x09 and move activation commands to **Application Group 0x08 - Manufacturer Configuration Messages**; Misc. clarifications and corrections

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 Technical Support 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 INFORMATION

This device complies with Part 15 of the FCC Rules. Operation 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.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment. RF Exposure: A distance of 20 cm shall be maintained between the antenna and users, and the transmitter may not be co-located with any other transmitter or antenna.

CUR/UR

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

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 Réglement 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 conformé 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 ***UL 60950-1, 2nd Edition, 2011-12-19*** (Information Technology Equipment - Safety - Part 1: General Requirements), ***CSA C22.2 No. 60950-1-07, 2nd Edition, 2011-12*** (Information Technology Equipment - Safety - Part 1: General Requirements).

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 (Pb).

1 Table of Contents

1 Introduction

1.1 About This Document

This document describes the master set of messages a host can send and receive via byte-by-byte direct communication with secure card reader authenticator devices that implement MagTek Common Message Format (MCMF), such as oDynamo (referred to in this document as "the device").

1.2 About Terminology

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

- **Device** refers to the MagTek product that receives and responds to the command set specified in this document. Devices include oDynamo.
- **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 BLE, 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 word **terminal** uses the EMV definition, which may mean a stationary interface for a cashier or teller at a point of sale or bank, an ATM or other unattended device, a handheld service interface on an air or water craft, and so on. In some situations the terminal interacts with the operator, though in self-service situations the terminal might interact with a cardholder directly.

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

1.3 About Connection Types

This device and related products use a common communication protocol across a variety of physical connection layers, which can include universal serial bus (USB), Ethernet, RS-232, and Bluetooth Low Energy (BLE). The set of available connection layers depends on the device. Details for communicating with devices via each physical connection type are provided in section **3 Connection Types**.

1.4 About Device Features

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

Table 1-1 - Device Features

Feature	oDynamo	Reserved	Reserved	Reserved	Reserved	Reserved
General Features						
Signature Capture ("SC")	No					
Custom messages	No					
Bitmaps	No					
Clear text user data	No					
Capacitive Keypad ("Cap Keypad")	No					
Activation codes						
Power management	No					
PCI 4.x Key Block	Yes					
IntelliHead	No					
Max financial card PAN length	19					
MagneSafe 2.0 (MagneSafe 2.0)	No					
Communication Interfaces						
USB Connection to peripherals	No					
USB Connection to host	Yes					
TCP/IP over 802.11 wireless connection	No					
Ethernet connection	Yes					
Apple 30-pin connection	No					
RS-232 connection	Yes					
Bluetooth connection ("BLE")	No					
SRED Options						
SRED	Yes					
Non-SRED	No					
EMV Features						

Feature	oDynamo	Reserved	Reserved	Reserved	Reserved	Reserved
Chip card contact	Yes					
Chip card **L1** mode	No					
Chip card **L2** mode	Yes					
RID CAPK Key Slots	Yes					
Multiple payment brand defaults	Yes					
Chip card contactless	No					
PayPass support	No					
payWave support	No					
Expresspay support	No					
D-PAS support	No					
Configurable EMV Support	No					

1.5 About APIs

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

2 About Messages, Commands, Responses, and Notifications

The host and the device communicate with each other by exchanging blocks of data called **Messages**, which can be either **Commands**, **Responses** to commands, or **Notifications**. For example, the host may send a *command* to the device to change a configuration setting, and the device may send a *response* that the command was successful; when a cardholder inserts a card, the device may send a *notification* to the host that a cardholder has initiated a transaction.

The device can only service one command at a time, and sends each response within a pre-determined finite amount of time after receiving the command. After sending a command, the host must wait until the device returns a response before sending another command.

The device sends notifications to the host if the device's state changes or if an external event occurs, such as a cardholder inserting a card. The device can send a notification at any time, and does not expect a response or any specific action from the host.

Although the wrappers for messages may be different depending on the connection type the host and device are using to communicate, the message payload format and contents are exactly the same. For example, apart from the wrapper differences defined in section **3 Connection Types**, a message sent over an RS-232 connection will be exactly the same as a message sent over a USB-HID connection.

2.1 Message Format

Messages exchanged between the host and the device consist of a required **header**, which consists of three or four specific tag-length-value (TLV) data objects, plus in many cases a fifth top-level TLV data object containing a data payload. **Table 2-1** shows the header TLV data objects in the order the host should use when sending messages, and should expect when receiving messages. Details about each of the top-level data objects are provided in the following sections, and every command and notification in section **4 Command Set** provides concrete usage tables that show byte-by-byte how the host software should compose or interpret the message.

Table 2-1 - Message Format

Tag	Data Value / Data Object(s)
C0	**Message Type Data Object (Tag C0),** sometimes abbreviated MTYP (always included)
C1	**Application ID Data Object (Tag C1)**, sometimes abbreviated APPID (always included)
C2	**Command ID Data Object (Tag C2)**, sometimes abbreviated CMDID (always included)
C3	**Result Code Data Object (Tag C3)**, sometimes abbreviated RC (not always included)
C4	**Data Field Data Object (Tag C4 or E0)**, sometimes abbreviated DATA (not always included)

The TLV data objects are constructed using the Basic Encoding Rules (BER) for the industry standard format defined in ***ITU-T X.680|ISO/IEC 8824-1*** and ***ITU-T X.690|ISO/IEC 8825-1***. These standards are also the basis of ***EMV Integrated Circuit Card Specifications for Payment Systems 4.3, Part IV, Annex B Rules for BER-TLV Data Objects***, so the latter can serve as a useful point of reference, especially for programmers who are familiar with the operation of chip cards. Summarizing those specifications, each TLV data object follows these basic rules:

- The **Tag** portion is a single byte that identifies the TLV data object [such as **0xC0** in the case of **Message Type Data Object (Tag C0)**].

- The **Length** portion is calculated as the total length of the Data portion that follows it. Lengths can be either one byte long from `0x00` to `0x7F`, or multiple bytes starting with `0x80` or higher, in which case the lower 7 bits of the first byte specify how many subsequent bytes will indicate the length of the Data portion that follows. For example, in the case of Length `8201C3`, `0x82` is greater than `0x7F`, and the lower 7 bits equal `0x02`, so the next 2 bytes `0x01C3` give the total length of the data block, 451 bytes.

- The **Data** portion is the actual payload of the TLV data object.

2.1.1 Message Type Data Object (Tag C0)

The **Message Type** TLV data object specifies the message type: Either **Command**, **Response**, or **Notification**. **Table 2-2** formally defines the TLV data object.

Table 2-2 -Message Type TLV Data Object (Tag C0)

TAG	1 Byte	C0	Message Type (MTYP) Tag
LEN	1 or Multi Byte	XX	For Primitive Devices Length is Always 1
DATA	Byte	Value	Definition
	1	01	Command message type
		02	Response message type
		03	Notification message type

For device operations that take long time or indefinite amount of time, the host usually sends a command that initiates the operation, and the device responds to indicate it has started the operation. When the operation completes, the device sends a Notification to the host.

2.1.2 Application ID Data Object (Tag C1)

The **Application ID** TLV data object specifies which Application ID the message belongs to. Each functional subsystem of the device has a unique Application ID, and contains a defined message set pertaining to that subsystem. For example, a device might contain an **MSR** application for its magnetic stripe reader and an **EMV L2 Contact** application for interacting with contact chip cards. **Table 2-3** formally defines the TLV data object.

Table 2-3 - Application ID TLV Data Object (Tag C1)

TAG	1 Byte	C1	Application ID (APPID)
LEN	1 or Multi Byte	XX	For Primitive Devices Length is Always 1
DATA	Byte	Value	Definition
	1	00..7F	This range is for generic messages. A generic message has a common message set for different device models. **Application Group 0x00 - Device Information Messages** **Application Group 0x01 - General Messages** **Application Group 0x02 - Authentication Messages** **Application Group 0x03 - Device Configuration Messages** **Application Group 0x04 - Magnetic Stripe Reader (MSR) Messages** **Application Group 0x05 - PAN Messages** **Application Group 0x07 - EMV L2 Contact Messages** (Chip Card L2 Mode Only) **Application Group 0x08 - Manufacturer Configuration Messages**
		80..FF	This range is for custom messages. A custom application has a unique message set for a particular device model.

2.1.3 Command ID Data Object (Tag C2)

The **Command ID** TLV data object has a different meaning for each of the possible values in the **Message Type Data Object (Tag C0)**:

- For **Commands**, the Command ID defines the operation to be carried out by the device.
- For **Responses**, the Command ID refers to the operation that was carried out by the device, and always contains the same value as the **Command ID Data Object (Tag C2)** from the originating command.
- For **Notifications**, the Command ID specifies the event that has occurred in the device.

Table 2-4 formally defines the TLV data object.

Table 2-4 - Command ID TLV Data Object (Tag C2)

TAG	1 Byte	C2	CMDID
LEN	1 or Multi Byte	XX	For Primitive Devices Length is Always 1
DATA	Byte	Value	Definition
	1 (Message ID)	00-7F	This range is for generic messages. A generic message has a common meaning for different applications. The existence of generic messages allows standardizing on commonly used messages across multiple applications. Applications are not required to implement all generic messages. Generic messages are defined further elsewhere in this document.
		80-FF	This range is for custom messages. A custom message has a unique meaning for a particular application. Custom messages are defined further elsewhere in this document.

2.1.4 Result Code Data Object (Tag C3)

The **Result Code** TLV data object has different meaning for each of the possible values in the **Message Type Data Object (Tag C0)**:

- For **Commands**, the host should not send this data object to the device.
- For **Responses**, the result code reports the result of the operation that was carried out by the device.
- For **Notifications**, the result code reports the result of the event that has occurred in the device.

Table 2-5 formally defines the TLV data object.

Table 2-5 - Result Code TLV Data Object (Tag C3)

TAG	1 Byte	C3	Result Code (RC)
LEN	1 or Multi Byte	XX	For Primitive Devices Length is Always 1
DATA	Byte	Value	Definition
	1 (Result Code)	00-7F	This range is for generic result codes. A generic result code has a common meaning for different applications. The following values are valid: 0x00 = OK / Done 0x01 = Failure 0x02 = Warning 0x03 = User Cancel 0x04 = Timeout 0x05 = Host Cancel 0x06 = Verify fail 0x07 = Bad Message Header 0x08 = Bad Application ID 0x09 = Bad Message ID 0x0A = Bad Parameter 0x0B = System State Error 0x0F = Current Device Status Prohibits Command 0x10 = Command not supported 0x11 = Requested item not available 0x12 = No card inserted 0x13 = Wrong card inserted 0x14 = Smart card not accessible 0x15 = Application already running 0x16 = Requested item expired 0x17 = Configuration locked, modification prohibited 0x18 = Error state 0xFF = Bad Message Format 0x10 = Keypad Security 0x11 = Calibration Done 0x12 = Write with duplicate RID and index 0x13 = Write with corrupted Key 0x14 = CA Public Key reached maximum capacity 0x15 = CA Public Key read with invalid RID or Index 0x15 - RID error / Index not found

		80-FF	This range is for custom result codes. A custom result code has a unique meaning for a particular application. 0x80 = Device Error - A device error or tamper has been detected, the device certificate is missing or has been changed, or signature is not correct. 0x81 = Device not Idle 0x82 = Data Error or Bad Parameter(s) - The command contains bad parameters. For example, in a big block message transfer, the parameters in any packet 1 through n don't match (or don't follow) the previous data packet's parameters; it may also indicate a bad CBC-MAC ACKSTS, wrong serial number, or a bad key. 0x83 = Length Error - The data size is 0 or is larger than the available buffer size, or a data packet is incomplete, or MagTek device OID of the certificate doesn't match the predefined OID 0x84 = PAN Exists 0x85 = No Key or Key is incorrect - No authentication key, no Acquirer Master Key, or an invalid key is found in the device 0x86 = Device busy 0x87 = Device Locked - More than 120 PINs were entered within one hour, or there have been three authentication failures, or a previous call to a command has locked the device's configuration 0x88 = Auth required 0x89 = Bad Auth - Host has sent an incorrect authentication token (e.g., the decrypted random token or device serial number doesn't match the device's current values) 0x8A = Device not Available, Device Status is not OK, Touchscreen is not connected or doesn't exist, or Authentication challenge token has timed out (i.e. is not used within 5 minutes) 0x8B = Amount Needed - If PIN amount is required, no amount has been sent 0x8C = Security Module error - If Security Module is missing or is not working correctly 0x90 = Cert non-exist - For unbind/rebind/key injection, the associated certificate doesn't exist 0x91 = Expired (Cert/CRL) 0x92 = Invalid (Cert/CRL/Message) 0x93 = Revoked (Cert/CRL) 0x94 = CRL doesn't exist 0x95 = Cert exists 0x96 = Duplicate KSN/Key

2.1.5 Data Field Data Object (Tag C4 or E0)

If there is additional data associated with the message, it is contained in the **Data Field** TLV data object. The length of this field is equal to the length of the whole message minus the length of the message header [**Message Type Data Object (Tag C0)**, **Application ID Data Object (Tag C1)**, **Command ID Data Object (Tag C2)**, and **Result Code Data Object (Tag C3)**].

For Primitive data, the Data Field data object is identified with Tag C4. C4 is only used when the data portion of the message only contains raw data and will not include any embedded TLV data objects. **Table 2-6** formally defines the C4 data object.

Table 2-6 - Primitive Data Field TLV Data Object (Tag C4)

TAG	C4	Primitive tag, for raw data with no encryption
LEN	XX	Length of Data
Data	Value (Hex)	Value of Data

For Constructed data, the Data Field data object is identified with Tag E0. E0 is used in two cases:

- If the data portion of the message wraps additional TLV data objects. The TLV data objects that can be embedded in the Data Field are detailed in section **2.2 Data Field**, or in the documentation in section **4 Command Set** for messages that use them.
- If the data portion of the message is encrypted. In this case, the Data Field will contain additional TLV data objects according to the device's encryption scheme.

In both cases, the data inside an E0 (Constructed type) Data Field is also encoded using the BER TLV data object rules explained in section **2.1 Message Format**. It becomes very important pay careful attention to the length of every data object to encode/decode correctly. Programmers way wish to recursively construct those data objects by "nesting" from the inside out.

Table 2-7 formally defines the E0 data object.

Table 2-7 - Constructed Data Field TLV Data Object (Tag E0)

TAG	E0	E0 = Constructed tag, meaning Data contains other tags
LEN	XX	Length of Data
Data	Value (Hex)	Other tags are present in data section

2.2 Data Field Content Objects

This section specifies the data objects that can be embedded in the **Data Field Data Object (Tag C4 or E0)** when it contains Constructed data tagged with type E0.

2.2.1 Data Object F1 - Device Status

The device uses a data object identified by tag F1 to send the device status:

Table 2-8 - Contents of Data Object F1 Device Status

Tag	Len	Value(s) / Description
F1	Calculated	DF51 < Device Status ><len><val> (see **Table 2-9**) DF52 < Device Certificate Status ><len><val> (see **Table 2-10**) DF53..DF5F <Reserved Status><len><val>

2.2.1.1 Data Object DF51 Device Status

Hardware errors will cause the device to not respond to any command, except for device status inquiry.

Table 2-9 - Contents of Data Object DF51 Device Status

Value	Bit 7	6	5	4	3	2	1	0
Byte 0 Hardware				UCI Error		TRNG Error		Crypto Engine Error
Byte 1 Reserved								
Byte 2 Reserved								
Byte 3 Reserved								
Byte 4 Security status					Real Time Clock Status Setting 0 = RTC has been set to current time. 1 = RTC has not been set to current time.	SYSTEM-CLEARED	Tamper occurred: 0 = Device not tampered 1 = Device tampered	Tamper activated: 0 = Activated 1 = Not activated
Byte 5 One-Time-Programmable (OTP) Memory Status							MAC Address Status: 0 = MAC has been written to OTP 1 = MAC has not been written to OTP	Serial Number Status: 0 = SN has been written to OTP 1 = SN has been NOT written to OTP
Byte 6 Reserved								

Value	Bit 7	6	5	4	3	2	1	0
Byte 7 Reserved								
Byte 8 Device Dismount Status						Device Removed: 0 = Device not in Removed state 1 = Device in Removed state	Device Installed: 0 = Device not in installed state 1 = Device in installed state	Device Pre-activated: 0 = Device not in preactivated state 1 = Device in preactivated state
Byte 9 Reserved								
Byte 10 Reserved								
Byte 11 Reserved								

2.2.1.2 Data Object DF52 Device Certificate & Key Status

Table 2-10 - Contents of Data Object DF52 Device Certificate & Key Status

Value	Bit 7	6	5	4	3	2	1	0
Byte 0 Device Certificate Status	PIN CRL	MSR Keyloader Cert	PIN Keyloader Cert	Device Cert	MSR SUB CA Cert	PIN SUB CA Cert	Device CA Cert	CA Unbind Cert
Byte 1 Device Certificate Status		TLS Cert						MSR CRL
Byte 2 Reserved								AES Key
Byte 3 Reserved								

2.2.2 Data Object F4 - Magnetic Stripe Reader Card Data

The device uses TLV Data Object F4 wrapped in **Data Object F9 - MACed Message** to transmit magnetic stripe reader data to the host, per **Table 2-11**.

Table 2-11 - Contents of TLV Data Object F4 – MSR Card Data

```
F9 TLV Container

     F4 TLV Container
          DFDF31<len><Masked T1 Data>
          DFDF33<len><Masked T2 Data>
          DFDF35<len><Masked T3 Data>
          DFDF36 - Track 1 Status (0x00 = OK, 0x01 = Empty, 0x02 =
Error, 0x03 = Disabled)
          DFDF38 - Track 2 Status
          DFDF3A - Track 3 Status
          DFDF43 - Magneprint Status
          DFDF4F - Encode type
          F8<len> /* container tag for encryption */
               DFDF59<len><encrypted data primitive@1, refer to notes
below>
               DFDF51<len><encryption type>
               DFDF56<len><KSN>
               DFDF58<len><val>(# of bytes of padding in DFDF59)
               DFDF25<len><Device Serial Number>

DFDF6C<len><MAC>


*****************************************************
Notes:
>>We are MACing the F9 container. The MAC length will be 4 bytes.

>><encrypted data container @1>

   FA <len>
              DF41 <len> Clear text Data for track 1
              DF42 <len> Clear text Data for track 2
              DF43 <len> Clear text Data for track 3
              DF44 - Magneprint - Cleartext Data

Notes for constructing the MSR data message:
The MSR data message has the following form:
*****************************************************************
**
C0 01 xx /*last byte depends on whether it's a response or
notification*/
C1 01 04
C2 01 12
C3 01 00 /* whether to include this TLV depends on message type
(response or notification)*/
```

```
E0  <E0 TLV len>  <F9 TLV for MACed MSR data>  <DFDF6C TLV for MAC
value>
**********************************************************************
**
The MAC value is calculated based on the padded <F9 TLV for MACed MSR
data> (the MACing operation requires MACed data length to be multiple
of 8). However, after MACing calculation, the padding bytes should not
be included in the F9 TLV. The receiving end is responsible for
padding the F9 TLV when verifying the corresponding MAC.
The MAC is stored in the DFDF6C TLV.
<E0 TLV len> is the totoal of the <F9 TLV for MACed MSR data>  and
<DFDF6C TLV for MAC value>.
```

Encryption Type data object DFDF51 uses one byte to represent the key scheme, encryption algorithm and variant. **Table 2-12** lists the possible values.

Table 2-12 - Contents of Encryption Type Data Object DFDF51

Tag	Description
DFDF51	MSR Encryption Type: 0xxx xxxx = Fixed key 1xxx xxxx = DUKPT key xx00 xxxx = TDES xx01 xxxx = AES xxxx xx00 = Data variant xxxx xx01 = PIN variant xxxx xx10 = MAC variant

2.2.3 Data Object F8 - Encrypted Data

TBD

2.2.4 Data Object F9 - MACed Message

TBD

3 Connection Types

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

3.1 How to Use Network Connections (Ethernet or 802.11 Wireless Only)

3.1.1 How to Use Ethernet Connections (Ethernet Only)

When the device is connected to a network via a 10/100 Ethernet port, it will attempt to contact a DHCP server to acquire a dynamic IP address during power-up. See the network administrator to determine the IP address the DHCP server assigned to the device. After determining the IP address, use **port 5000** to communicate with the device.

3.1.2 How to Send Commands Using the Network Connection

The messages exchanged between the host and the device on the TCP/IP connection do not require any wrappers or encoding: The binary data flowing through the connection is identical to the message format defined in section **2 About Messages, Commands, Responses, and Notifications** and in section **4 Command Set**. A **C0** at the beginning of a message's usage table means that without any preparation other than binding to the port, the host begins sending the command by sending a single binary byte `0xC0`.

For example, to send **Command 0x00:0x10 - Get Product ID** to the device over TCP/IP, the host should send a stream consisting of single binary byte `0xC0`, followed by single binary byte `0x01`, followed by single binary byte `0x01`, followed by single binary byte `0xC1`, and so on following the sequence in **Table 4-3**. The device will then send a response over the same connection according to the sequence in **Table 4-4**.

3.2 How to Use RS-232 Connections (RS-232 Only)

The messages exchanged between the host and the device on the RS-232 connection require a small wrapper and must be ASCII encoded hexadecimal (‘0’ through ‘F’ only). The message format is defined in section **2 About Messages, Commands, Responses, and Notifications** and in section **4 Command Set**. ASCII encoding means when the device intends to send **C0** at the beginning of a message’s usage table, it should send two ASCII bytes, where ‘C’ is ASCII 0x43 and ‘0’ is ASCII 0x30. When the device is using default settings, the host should send a line feed (0x0A) to signal the end of the message. The device’s responses and notifications will be wrapped and encoded the same way.

For example, to send **Command 0x00:0x10 - Get Product ID** to the device over the RS-232 connection, the host should send a stream consisting of ASCII ‘C’ (0x43), ASCII ‘0’ (0x30), ASCII ‘0’ (0x30), ASCII ‘1’ (0x31), ASCII ‘0’ (0x30), ASCII ‘1’ (0x31), ASCII ‘C’ (0x43), ASCII ‘1’ (0x31), and so on following the sequence in **Table 4-3**, then a line feed (0x0A). The device will then send a response over the same connection according to the sequence in **Table 4-4**, then a line feed (0x0A).

The devices only use **TXD** and **RXD**; hardware handshaking is not available. The default serial settings are **9600 bps**, **No parity**, **8 data bits**, and **1 stop bit**.

The device can optionally be configured by the manufacturer to expect / transmit a Starting Byte and an ASCII CRC checksum in this order: <Starting Byte> <Message in ASCII> <CRC in ASCII> <Ending Byte>. The device’s default setting is “raw mode,” where the device only expects / transmits the Ending Byte, which is set to Line Feed (0x0A). Other popular choices when ordering a device may include Starting Byte=STX (0x02); Ending Byte=ETX (0x03) or CR (0x0D).

3.3 How to Use USB Connections (USB Only)

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

Developers can easily create custom software to communicate with the device using any framework that can communicate with a USB port. For example, developers can use the standard Windows USB HID driver with Visual Basic or Visual C++. MagTek has developed demonstration software that communicates with the device via this method, and developers can use it to test the device and to provide a starting point for developing other software. Because the device's USB implementation is operating system agnostic, the device can be used with other platforms as well, such as Linux. For more information, see the MagTek web site, or contact your reseller or MagTek Support Services.

The device is a full speed high-powered USB device that identifies itself with vendor **ID 0x0801** and product **ID 0x001B**. The device does not draw power from the USB port, and does not support USB Suspend or remote wakeup.

This device has programmable configuration properties stored in non-volatile memory. The properties can be configured at the factory, by the key loader, or by the end user. More details can be found in section **4 Command Set** in this document, and in a separate document which provides details about key loading.

3.3.1 About HID Usages

3.3.1.1 About Reports

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

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

- **Feature Reports**, which can be further divided into **Get** types and **Set** types. The host exclusively uses this type of report to send messages to the device.
- **Input Reports** are used by the device to send asynchronous responses or notifications to the host when a related feature report completes, or automatically when the device's state changes. This is common when an operation depends on cardholder action.
- **Output Reports**. Output reports are part of the USB HID standard, but are not used by this device.

3.3.1.2 About the Report Descriptor

The list of the device's available reports and their structure is sent to the host in a **report descriptor**, usually just after the device is connected to the USB port. Generally the details of the report descriptor are abstracted by the developer's HID API; however, should it become necessary to examine a report descriptor byte-by-byte, a full inventory of the report descriptor for these devices is provided in **Table 3-1**.

Table 3-1 - USB HID Report Descriptor

Item Tag (Value)	Raw Data
Usage Page (Vendor-Defined 33)	06 20 FF
Usage (Vendor-Defined 1)	09 01
Collection (Application)	A1 01
Report Size (8)	75 08
Logical Minimum (0)	15 00
Logical Maximum (255)	26 FF 00
Report ID (5)	85 05
Usage (Vendor-Defined 5)	09 05
Report Count (63)	95 3F
Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf)	B2 02 01
Report ID (2)	85 02
Usage (Vendor-Defined 32)	09 20
Report Count (63)	95 3F
Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Buf)	82 02 01
Report ID (3)	85 03
Usage (Vendor-Defined 33)	09 21
Report Count (5)	95 05
Output (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Bit)	91 02
End Collection	C0

3.3.2 How to Send Commands Using USB HID

The general sequence the host should use to send a message to the device is as follows:

1) The host sends a Set Feature Report with **Report ID 0x05**, containing the command message in binary format (bytes) as data.
2) The device asynchronously sends an Input Report with **Report ID 0x02**, containing the defined response to the originating command in binary format (bytes) as data. Depending on the command, the response may or may not contain a **Data Field Data Object (Tag C4 or E0)**.

The general sequence the device uses to send a message to the host is as follows:

1) The device's state changes, a cardholder or operator takes action, or new information becomes available.
2) The device asynchronously sends an Input Report with **Report ID 0x02**, containing the notification message in binary format (bytes) as data.

4 Command Set

This section documents the full set of messages that can be exchanged between the device and the host.

4.1 About Responses

4.1.1 ACK Response

After receiving a command that does not require the device to return data, the device sends a simple response to the host to acknowledge ("ACK") the command. The response includes the **Message Type Data Object (Tag C0)** indicating it is a response, the **Application ID Data Object (Tag C1)** and the **Command ID Data Object (Tag C2)** identical to the command the device is acknowledging, and a **Result Code Data Object (Tag C3)** reporting the results of the command.

Table 4-1 shows an example of an ACK Response from the device after the host has sent **Command 0x01::0x02 - Clear Session**, where the device is reporting "OK / Done."

Table 4-1 – Example ACK Response "OK / Done"

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = Response
C1	01	01	**Application ID Data Object (Tag C1)** = **Application Group 0x01 - General Messages**
C2	01	02	**Command ID Data Object (Tag C2)** = **Command 0x01::0x02 - Clear Session**
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done

Table 4-2 shows an example of an ACK Response from the device where the host has sent a command using an invalid message format (protocol violation) and the message is "Bad Message Format."

Table 4-2 -Example ACK Response "Bad Message Format"

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = Response
C1	01	10	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	10	**Command ID Data Object (Tag C2)** = **Command 0x01::0x10 - Send Big Block Command**
C3	01	FF	**Result Code Data Object (Tag C3) =** Bad Message Format

4.2 About Big Block Commands

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

For commands that require big block messages, the host should first compose the full command message, in local memory, according to the usage table of the desired command. It should then use **Command 0x01::0x10 - Send Big Block Command** to transmit the command to the device.

4.3 Application Group 0x00 - Device Information Messages

4.3.1 Command 0x00:0x10 - Get Product ID

The host uses this command to get the device's product ID.

Table 4-3 - Message Structure for Command 0x00:0x10 - Get Product ID

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	10	**Command ID Data Object (Tag C2)** = 0x10 Get Product ID

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-4 - Response to Command 0x00:0x10 - Get Product ID

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	10	**Command ID Data Object (Tag C2)** = 0x10 Get Product ID
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Product ID, e.g., 5999	

4.3.2 Command 0x00::0x12 - Get Capability String

The host uses this command to get the device's Capability String.

Table 4-5 - Message Structure for Command 0x00::0x12 - Get Capability String

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	12	**Command ID Data Object (Tag C2)** = 0x12 Get Capability String

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-6 - Response to Command 0x00::0x12 - Get Capability String

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	12	**Command ID Data Object (Tag C2)** = 0x12 Get Capability String
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Null-terminated string similar to "V=1,SC=1,SR=1,TR=1,MS2=1,PFK=1,UDE=2,CE=2,CLE=1,DR=1" where each of the comma-separated name-value pairs represents a device capability: V = Device Version SC = Signature Capture Support, 1=Supported, 0=Not Supported SR = SRED, 1=SRED, 0=NON-SRED TR = Token Reversal Support, 1=Supported, 0=Not Supported MS2 = MagneSafe 2.0 Support, 1=Supported, 0=Not Supported PFK = PIN Fixed Key Support, 1=Supported, 0=Not Supported UDE = User Data Entry Mode, 1=Encrypted Only, 2=Clear Text and Encrypted CE = Contact EMV Level Support, 1=L1, 2=L2 CLE = Contactless EMV Level Support, 1=L1, 2=L2 DR = Delayed Response Support, 1=Supported, 0=Not Supported	

4.3.3 Command 0x00::0x13 - Get Manufacturer

The host uses this command to get the device manufacturer.

Table 4-7 - Message Structure for Command 0x00::0x13 - Get Manufacturer

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	13	**Command ID Data Object (Tag C2)** = 0x13 Get Manufacturer

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-8 - Response to Command 0x00::0x13 - Get Manufacturer

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	13	**Command ID Data Object (Tag C2)** = 0x13 Get Manufacturer
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Manufacturer, e.g., MagTek, Inc.	

4.3.4 Command 0x00::0x14 - Get Product Name

The host uses this command to get the device's product name.

Table 4-9 - Message Structure for Command 0x00::0x14 - Get Product Name

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	14	**Command ID Data Object (Tag C2)** = 0x14 Get Product Name

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-10 - Response to Command 0x00::0x14 - Get Product Name

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	14	**Command ID Data Object (Tag C2)** = 0x14 Get Product Name
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Product Name, e.g., oDynamo	

4.3.5 Command 0x00::0x15 - Get Secure Tracking Number

The host uses this command to get the device's secure tracking number.

Table 4-11 - Message Structure for Command 0x00::0x15 - Get Secure Tracking Number

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	15	**Command ID Data Object (Tag C2)** = 0x15 Get Secure Tracking Number

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-12 - Response to Command 0x00::0x15 - Get Secure Tracking Number

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	15	**Command ID Data Object (Tag C2)** = 0x15 Get Secure Tracking Number
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	26	**Data Field Data Object (Tag C4 or E0)** = Bytes 0..25 Secure Tracking Number ASCII string. Example "B559846789012345678901234 56"	

4.3.6 Command 0x00::0x18 - Get Network Information (Ethernet Only)

The host uses this command to get information about the device's network connection.

Table 4-13 - Message Structure for Command 0x00::0x18 - Get Network Information (Ethernet Only)

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	18	**Command ID Data Object (Tag C2)** = 0x18 Get Network Information
C4	01	**Data Field Data Object (Tag C4 or E0)** = Type of Information 0x00 = Request Ethernet MAC Address 0x01 = Request Ethernet IP Address	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-14 - Response to Command 0x00::0x18 - Get Network Information (Ethernet Only)

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	18	**Command ID Data Object (Tag C2)** = 0x18 Get Network Informations
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Type of Information 0x00 = Ethernet MAC Address 0x01 = Ethernet IP Address Bytes 1..n.Network Information String MAC Address in ASCII format, e.g., 112233445566 or IP Address e.g., 111.222.333.444	

4.3.7 Command 0x00::0x26 - Get Core Firmware Build Info

The host uses this command to get the buid info for the device's core firmware.

Table 4-15 - Message Structure for Command 0x00::0x26 - Get Core Firmware Build Info

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	26	**Command ID Data Object (Tag C2)** = 0x26 Get Core Firmware Build Info

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-16 - Response to Command 0x00::0x26 - Get Core Firmware Build Info

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	26	**Command ID Data Object (Tag C2)** = 0x26 Get Core Firmware Build Info
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = ASCII string containing "[version name][YYYY-MM-DD]HH:MM:SS"	

4.3.8 Command 0x00::0x27 - Get CT-L2 Version

The host uses this command to get the device's EMV L2 version, if any.

Table 4-17 - Message Structure for Command 0x00::0x27 - Get CT-L2 Version

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	27	**Command ID Data Object (Tag C2)** = 0x27 Get CT-L2 Version

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-18 - Response to Command 0x00::0x27 - Get CT-L2 Version

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	27	**Command ID Data Object (Tag C2)** = 0x27 Get CT-L2 Version
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = ASCII string containing a "[version name][YYYY-MM-DD]HH:MM:SS"	

4.3.9 Command 0x00::0x28 - Get Serial Number

The host uses this command to get the device's serial number.

Table 4-19 - Message Structure for Command 0x00::0x28 - Get Serial Number

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	28	**Command ID Data Object (Tag C2)** = 0x28 Get Serial Number

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-20 - Response to Command 0x00::0x28 - Get Serial Number

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	00	**Application ID Data Object (Tag C1)** = 0x00 Device Information
C2	01	28	**Command ID Data Object (Tag C2)** = 0x28 Get Serial Number
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	16	**Data Field Data Object (Tag C4 or E0)** = Serial Number in ASCII starting with 1111, e.g., "1111xxxxxxxxxxx"	

4.4 Application Group 0x01 - General Messages

4.4.1 Command 0x01::0x02 - Clear Session

The host uses this command to direct the device to clear all existing session data, including account data, encrypted PIN block, PAN, and amount, and return to the idle state.

Table 4-21 - Message Structure for Command 0x01::0x02 - Clear Session

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	02	**Command ID Data Object (Tag C2)** = 0x02 Clear Session

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-22 - Response to Command 0x01::0x02 - Clear Session

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	02	**Command ID Data Object (Tag C2)** = 0x02 Clear Session
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done

4.4.2 Command 0x01::0xFF - Device Reset

The host uses this command to direct the device to perform a reset.

Table 4-23 - Message Structure for Command 0x01::0xFF - Device Reset

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	FF	**Command ID Data Object (Tag C2)** = 0xFF Device Reset
C4	01	**Data Field Data Object (Tag C4 or E0)** = 00 = Soft Reset 01 = Hard Reset	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-24 - Response to Command 0x01::0xFF - Device Reset

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	FF	**Command ID Data Object (Tag C2)** = 0xFF Device Reset
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done

4.4.3 Notification 0x01::0xFF - Device Reset

The device sends this notification to the host to no it that it is about to perform a periodic automatic device reset, which it initiates after 23 hours of continuous operation to fulfill PCI requirements. MagTek recommends the host software pre-empt these automatic resets by initiating resets in advance of the 23 hour schedule, at times that are least disruptive to the solution design.

Table 4-25 - Response to Notification 0x01::0xFF - Device Reset

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	FF	**Command ID Data Object (Tag C2)** = 0xFF Device Reset

4.4.4 Command 0x01::0x04 - Get Device Status

The host uses this command to request the device status, such as Session State, Device State, and Status.

Table 4-26 - Message Structure for Command 0x01::0x04 - Get Device Status

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 Get Device Status

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-27 - Response to Command 0x01::0x04 - Get Device Status

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 Get Device Status
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done
E0	Calculated	**Data Field Data Object (Tag C4 or E0)** = **Data Object F1 - Device Status**	

4.4.5 Notification 0x01::0x04 - Send Device Status

In addition to sending device status in response to **Command 0x01::0x04 - Get Device Status**, the device will automatically send the same data when the device powers up, restarts, or changes state.

Table 4-28 - Message Structure for Notification 0x01::0x04 - Send Device Status

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 Get Device Status
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done
E0	Calculated	**Data Field Data Object (Tag C4 or E0)** = **Data Object F1 - Device Status**	

4.4.6 Command 0x01::0x10 - Send Big Block Command

The host uses this command to send command messages to the device as a sequence of packets. The host should follow this sequence:

1) In local memory, compose the entire command message as documented by the desired command's usage table (for example, **Command 0x01::0x17 - Update Firmware**).
2) Calculate the length of the fully-composed command message. Divide the fully-composed command message into packets that are shorter than the connection type's maximum packet size.
3) Send command 0x01::0x10 to the device as "Packet 0." Packet 0 is short and sets up how much data the device should expect across the whole big block operation. The C4 data object can be of varying length, and its value should be a 2-byte Packet Number equal to 00 00, plus a 2-byte Packet Length, plus Packet Data containing the Total Command Message Length, in bytes, the device should expect to receive when concatenating all subsequent packets (see **Table 4-29**).
4) Wait to receive a response from the device (see **For every** packet the host sends, if an error occurs (such as an out of order packet), the device will terminate the command, report the error using an **ACK Response** containing the result code, and will stop expecting to receive subsequent big block packets. The host should stop sending them. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**.

If no error occurs, the device will respond to every packet as follows:

5) Table 4-31).
6) Continue sending Command 0x01::0x10 to send Packets 1 through n, incrementing the Packet Number by 1 each time, until the host has sent the fully-composed command message. After sending each packet, wait for the device's response to reduce risk of packets arriving out of order. The C4 data object for packets 1 through n can be of varying length, and its value should be a 2-byte Packet Number that increments with each call to this command, plus a 2-byte Packet Data Length, plus the Packet Data (see **Table 4-30**).
7) Listen for a final response from the device acknowledging the completed command. This means after sending the final packet, the host should expect to receive two responses: One for the final packet sent with Command 0x01::0x10, and one after the device has finished processing all the uploaded data and responds to the fully-composed command message the host has sent (for example, **Command 0x01::0x17 - Update Firmware**).

Table 4-29 - Message Structure for Command 0x01::0x10 - Send Big Block Command (Packet 0 Only)

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	10	**Command ID Data Object (Tag C2)** = 0x10 Send Big Block Command Message
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = 2-byte Packet Number = 00 00 2-byte Packet Data Length, LSB first Packet Data = Total Command Message Length in bytes the device is sending using big block packets, LSB first	

Table 4-30 - Message Structure for Command 0x01::0x10 - Send Big Block Command (Packets 1 through n)

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	10	**Command ID Data Object (Tag C2)** = 0x10 Send Big Block Command Message
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = 2-byte Packet Number, LSB first 2-byte Packet Data Length, LSB first Packet Data, LSB first	

For every packet the host sends, if an error occurs (such as an out of order packet), the device will terminate the command, report the error using an **ACK Response** containing the result code, and will stop expecting to receive subsequent big block packets. The host should stop sending them. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**.

If no error occurs, the device will respond to every packet as follows:

Table 4-31 - Response to Command 0x01::0x10 - Send Big Block Command

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	10	**Command ID Data Object (Tag C2)** = 0x10 Send Big Block Command Message
C3	01	00	**Result Code Data Object (Tag C3) =** OK / Done

4.4.7 Command 0x01::0x17 - Update Firmware

The host uses this command to update the device's firmware. The host should first compose the entire message for this command, then transmit the entire message to the device in packets using **Command 0x01::0x10 - Send Big Block Command.**

Table 4-32 - Message Structure for Command 0x01::0x17 - Update Firmware

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	17	**Command ID Data Object (Tag C2)** = 0x17 Update Firmware
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 1 Subcommand 0x01 = Update boot loader 0x02 = Reserved 0x03 = Update firmware image 0x04 = Reserved Bytes 2..n: Firmware image binary data	

After the device receives the final packet of this command, the device will validate the firmware image, and if the validation passes, it will commit the image to the selected firmware storage location. In all cases, the device will also send an additional response:

Table 4-33 - Response to Command 0x01::0x17 - Update Firmware

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	17	**Command ID Data Object (Tag C2)** = 0x17 Update Firmware
C3	01	**Result Code Data Object (Tag C3)**: 0x00 = Update successful 0x01 = Invalid firmware image loaded in buffer 0x02 = Signature error 0x03 = Build version error 0x09 = Other error	

Although this command works on all connection types, for speed reasons, MagTek recommends using an Ethernet or USB connection to upgrade the device's firmware.

4.4.8 Notification 0x01::0x40 - Card Inserted / Identified / Removed

The device uses this notification to signal to the host that a cardholder has inserted or removed a card, and to identify the type of card. Upon insertion or removal of a card, the device sends this notification to provide a Card Present Indication (CPI). After the device finishes identifying the type of card, it sends this notification again to provide an ICC Present Indication (IPI). When the CPI indicates a card is present and the IPI reports it is a chip card, the host is free to begin an EMV transaction, or in the case of magnetic stripe cards, to prompt the cardholder to remove the card to swipe on exit.

Table 4-34 - Message Structure for Notification 0x01::0x40 - Card Inserted / Identified / Removed

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	40	**Command ID Data Object (Tag C2)** = 0x40 Card Inserted/Identified/Removed
C4	02	**Data Field Data Object (Tag C4 or E0)** = Byte 0: Card Present Indication (CPI) 0x00 = Card was present and has been removed 0x01 = Card was not present and is now present Byte 1: Card Type Present Indication (IPI) 0x00 = Card is not a chip card (ICC card) 0x01 = Card is a chip card (ICC card) 0x02 = Future proof card detected	

4.4.9 Command 0x01::0x50 - Subscribe to Notifications

Table 4-35 - Message Structure for Command 0x01::0x50 - Subscribe to Notifications

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	50	**Command ID Data Object (Tag C2)** = 0x50 Subscribe to Notifications
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Subscribe or Unsubscribe 0x01 = Subscribe 0x02 = Unsubscribe Byte 1 Messages to Subscribe To 0x00 = Subscribe/unsubscribe to all notifications 0x01 = Subscribe/unsubscribe to specific notifications If Messages to Subscribe To is 0x01: Bytes 2..n contain the list of notification IDs to subscribe to	

4.4.10 Command 0x01::0x53 - Activate Device

The host uses this command to set the device to the **installed** state. The device must be in the **pre-activated** state before it will accept this command. To read the device's current state, use **Command 0x01::0x04 - Get Device Status**.

Table 4-36 - Message Structure for Command 0x01::0x53 - Activate Device

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	53	**Command ID Data Object (Tag C2)** = 0x53 Activate Device
C4	06	**Data Field Data Object (Tag C4 or E0)** = 6-digit TECH ID	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-37 - Response to Command 0x01::0x53 - Activate Device

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	01	**Application ID Data Object (Tag C1)** = 0x01 General Messages
C2	01	53	**Command ID Data Object (Tag C2)** = 0x53 Activate Device
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done

4.5 Application Group 0x02 - Authentication Messages

4.5.1 Command 0x02::0x0B - Get Challenge

The host uses this command to get challenge information from the device.

Table 4-38 - Message Structure for Command 0x02::0x0B - Get Challenge

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	02	**Application ID Data Object (Tag C1)** = 0x02 Authentication Messages
C2	01	0B	**Command ID Data Object (Tag C2)** = 0x0B Get Challenge
C4	02	**Data Field Data Object (Tag C4 or E0)** = 0xDF70 = PIN key 0xDF71 = MSR key 0xDF72 = PIN certificate 0xDF73 = MSR certificate 0xDF74 = Device Authentication signed by PIN cert 0xDF75 = Device Authentication signed by MSR cert 0xDF76 = Inject Fixed PIN key signed by PIN cert 0xDF78 = Inject Authentication key signed by PIN cert 0xDF79 = Inject Authentication key signed by MSR cert 0xDF7A = Inject Configuration signed by PIN cert 0xDF7B = Inject Configuration signed by MSR cert 0xDF7C = MFG command 0xDF7D = Authentication	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-39 - Response to Command 0x02::0x0B - Get Challenge

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	02	**Application ID Data Object (Tag C1)** = 0x02 Authentication Messages
C2	01	0B	**Command ID Data Object (Tag C2)** = 0x0B Get Challenge
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
E0	Calculated	**Data Field Data Object (Tag C4 or E0)** = Bytes 0..1 Key ID: DF70 = PIN key DF71 = MSR key DF72 = PIN Cert DF73 = MSR Cert DF74 = Device Authentication signed by PIN cert DF75 = Device Authentication signed by MSR cert DF76 = Inject Fixed PIN key signed by PIN cert DF78 = Inject Authentication key signed by PIN cert	

Tag	Len	Value(s) / Description
		DF79 = Inject Authentication key signed by MSR cert DF7A = Inject Configuration signed by PIN cert DF7B = Inject Configuration signed by MSR cert DF7C = Authentication DF7D = MFG command Bytes 2..13 Data Block: If Key_ID <= 0xDF7C or Key_ID = 0xDF64: Bytes 2..9 contain the device serial number Bytes 10..13 contain the random token

4.5.2 Command 0x02::0x0E - Get Key Information

The host uses this command to get key information from the device.

Table 4-40 - Message Structure for Command 0x02::0x0E - Get Key Information

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	02	**Application ID Data Object (Tag C1)** = 0x02 Authentication Messages
C2	01	0E	**Command ID Data Object (Tag C2)** = 0x0E Get Key Information
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Info ID from **Table 4-42**	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-41 - Response to Command 0x02::0x0E - Get Key Information

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	02	**Application ID Data Object (Tag C1)** = 0x02 Authentication Messages
C2	01	0E	**Command ID Data Object (Tag C2)** = 0x0E Get Key Information
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Info ID from **Table 4-42** Byte 1 Key Status, if Info ID < 0x80 0x00 = Empty (default) 0x01 = OK 0x02 = Exhausted Key Status, if Info ID = 0x80: 0x00 to 0x05 = KCV type from **Table 4-42** Byte 2 Data Length (Default=0x00, see **Table 4-42**) Bytes 3.. Data per **Table 4-42**	

Table 4-42 - Table of Info IDs and Data

Info ID	Key Status	Data length	Data	Description
0x00	1	Label length	AMK (Acquirer Master Key) label	If AMK (Acquirer Master Key) exists
0x01,0x02	2	20	KSN	If no more keys
0x01	1	20	KSN	PIN key

Info ID	Key Status	Data length	Data	Description
0x02	1	20	KSN	MSR key
0x03	1	calculated (<=59)	SN & subject's DN**	If PIN cert exists
0x04	1	calculated (<=59)	SN & subject's DN**	If MSR cert exists

*: lbllen = auth key's label length
**: SN = serial number of cert
DN = distinguished names of subject or issuer of cert
Data length varies with SN and DN length; max length is 59
***: its corresponding CA cert
****: KCV = Key Check Value, where the lowest 6 digits are valid

4.6 Application Group 0x03 - Device Configuration Messages

The host uses commands in this application to get and set the configuration of the device. Every configuration setting has a command to get the setting and a command to change the setting. When using get commands, the host should not include **Data Field Data Object (Tag C4 or E0)**. The device will respond with the current configuration values in **Data Field Data Object (Tag C4 or E0)**.

Caution
Locking can not be undone without returning the device to the supplier or manufacturer.

When the configuration is unlocked, the host can only change:

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

When the Config is set to locked, the host can not change any of the device configuration settings.

The host must send **Command 0x01::0xFF - Device Reset** to reset the device before the new EMV Mode control setting will take effect.

4.6.1 Command 0x03::0x60 - Set/Get Ethernet Configuration (Ethernet Only)

The host uses this command and its subcommands to set and get configuration settings for the device's Ethernet connection.

To change the device's Ethernet configuration, the host should follow these steps:

1) Call the command with the Set Ethernet IP Address Mode subcommand to select DHCP or Static.
2) If using Static, call the command with each Set subcommand for all remaining settings. If the host selected DHCP, the device ignores the remaining settings so further Set calls are unnecessary.
3) Call the command again with the Apply Changes subcommand. The device will begin using the new configuration immediately, and the settings will persist through subsequent power cycles and restarts.

To get information about the device's Ethernet configuration, the host should call this command with one of the Get subcommands, and interpret the device's response based on which subcommand it used.

Table 4-43 - Message Structure for Command 0x03::0x60 - Set/Get Ethernet Configuration (Ethernet Only)

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	60	**Command ID Data Object (Tag C2)** = 0x60 Set/Get Ethernet Configuration
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Subcommand. Set Commands are below 0x80, Gets are 0x80 and above. 0x00 = Apply Changes 0x01 = Set Ethernet IP Address (only if Ethernet IP Address Mode is set to Static) 0x02 = Reserved 0x03 = Set Ethernet IP Address Mode (DHCP vs. Static) 0x04 = Set Ethernet Gateway 0x05 = Set Ethernet Netmask 0x80 = Get All Ethernet Information 0x81 = Get Ethernet IP Address 0x82 = Get Ethernet MAC Address 0x83 = Get Ethernet IP Address Mode (DHCP vs. Static) 0x84 = Get Ethernet Gateway 0x85 = Get Ethernet Netmask Bytes 1..n.Network Configuration Data depends on which subcommand the host selected in the Subcommand byte. Apply Changes uses 0 bytes Set Static IP Address uses 4 bytes MSB first, e.g., 0xAABBCCDD Set IP Address Mode uses 1 byte, 0x00 = DHCP, 0x01 = Static Set Gateway uses 4 bytes MSB first e.g., 0xAABBCCDD Set Netmask uses 4 bytes MSB first e.g., 0xAABBCCDD	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-44 - Response to Command 0x03::0x60 - Set/Get Ethernet Configuration (Ethernet Only)

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	60	**Command ID Data Object (Tag C2)** = 0x60 Set/Get Ethernet Configuration
C3	01	**Result Code Data Object (Tag C3) =** 0x00 = OK / Done 0xFE = Invalid IP address 0xFD = Invalid Netmask 0xFC = Invalid Gateway address	
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Subcommand. Set Commands are below 0x80, Gets are 0x80 and above. 0x00 = Apply Changes 0x01 = Set Ethernet IP Address (only if Ethernet IP Address Mode is set to Static) 0x02 = Reserved 0x03 = Set Ethernet IP Address Mode (DHCP vs. Static) 0x04 = Set Ethernet Gateway 0x05 = Set Ethernet Netmask 0x80 = Reserved 0x81 = Get Ethernet IP Address 0x82 = Get Ethernet MAC Address 0x83 = Get Ethernet IP Address Mode (DHCP vs. Static) 0x84 = Get Ethernet Gateway 0x85 = Get Ethernet Netmask Bytes 1..n.Network Configuration Data depends on which subcommand the host selected in the Subcommand byte. Set commands use 1 byte equal to 0x00 Get MAC Address uses 6 bytes MSB first, e.g., 0xAABBCCDDEEFF Get IP Address use 4 bytes MSB first, e.g., 0xAABBCCDD Get IP Address Mode use 1 byte, 0x00 = DHCP, 0x01 = Static Get Gateway use 4 bytes MSB first e.g., 0xAABBCCDD Get Netmask use 4 bytes MSB first e.g., 0xAABBCCDD	

4.6.2 Command 0x03::0x71 - Set Device Configuration

The host uses this command to set the device configuration.

Table 4-45 - Message Structure for Command 0x03::0x71 - Set Device Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	71	**Command ID Data Object (Tag C2)** = 0x71 Set Device Configuration.
E0	Calculated	**Data Field Data Object (Tag C4 or E0)** = Recursive TLV container with a variable number of configuration items in primitive TLV format. See **Table 4-46**.	

Table 4-46 - Container E0 Tags, Lengths, and Values for Command 0x03::0x71 - Set Device Configuration

Tag	Len	Value(s) / Description
DFDFDF11	1	MSR encryption variant: 0 = Data variant 1 = PIN variant)
DFDFDF12	1	EMV L2 mode (enable or disable)
DFDFDF14	1	MSR nonstandard ISO decode enable: 0 = Disable non-standard ISO decoding 1 = Enable nonstandard ISO decoding (default) Standard ISO encoding is 7-bit ISO encoding on track 1 and 5-bit ISO encoding on tracks 2 and 3. Financial cards use standard ISO encoding. Nonstandard ISO encoding is considered any other combination of 7-bit ISO or 5-bit ISO encoding on any track. AAMVA encoding is also considered nonstandard ISO encoding because it is encoded as 7-bit ISO on track 1, 5-bit ISO on track 2, and 7-bit ISO on track 3.
DFDFDF15	1	MSR Track 1 Enable / Disable 0x00 = Disable 0x01 = Enable
DFDFDF16	1	MSR Track 2 Enable / Disable 0x00 = Disable 0x01 = Enable
DFDFDF17	1	MSR Track 3 Enable / Disable 0x00 = Disable 0x01 = Enable
DFDFDF18	1	MSR mask character (any printable ASCII character, typically set to “0” or “*”)

Tag	Len	Value(s) / Description
DFDFDF19	1	MSR number of leading unmasked digits (0 to 6)
DFDFDF1A	1	MSR number of trailing unmasked digits (0 to 4)
DFDFDF1C	1	Performance test
DFDFDF27	1	RS-232 CRC setting 0x00 = Do not include CRC 0x01 = Include CRC
DFDFDF28	1	RS-232 starting character
DFDFDF29	1	RS-232 ending character
DFDFDF30	1	Test Allowed 0x00 = Disable 0x01 = Enable
DFDFDF31	1	Configuration lock 0x00 = Unlock 0x01 = Lock
DFDFDF32	1	MSR Mask Check Digit Correction 0x00 = Disable 0x01 = Enable (default) When enabled, the device masks the PAN with ASCII “0” regardless of the MSR mask character setting, and one mask digit will be modified so the PAN check digit is correct.

4.6.3 Command 0x03::0x72 - Get Device Configuration

The host uses this command to get the device's communication interface configuration.

Table 4-47 - Message Structure for Command 0x03::0x72 - Get Device Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	72	**Command ID Data Object (Tag C2)** = 0x72 retrieve communication Interface Configuration.
C4	04	**Data Field Data Object (Tag C4 or E0)** = DFDFDFXX Tag associated with the targeted configuration item. See **Table 4-46** on page **62**.	

Table 4-48 - Response to Command 0x03::0x72 - Get Device Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	72	**Command ID Data Object (Tag C2)** = 0x72 Get Device Configuration
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	02	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Status 0x00 = Requested value not found 0x01 = Requested value found Byte 1 Requested configuration value. See **Table 4-46**.	

4.6.4 Command 0x03::0x80 - Get PAN / MSR Whitelist

The first 6 digits of a card number are known as the Issuer Identification Number (IIN), previously known as bank identification number (BIN). These identify the institution that issued the card to the cardholder.

The host uses this command to read the device's PAN Whitelist or MSR Whitelist, which are loaded securely by the manufacturer. Whitelisting allows the device to relax security for cards with specific IINs (e.g., loyalty cards). The MSR whitelist allows the device to send data from matching cards unencrypted. The PAN whitelist allows the device to make the whole PAN or a portion of the PAN available, which is typically needed by an external encrypting PIN pad (EPP) as part of creating an encrypted PIN block.

All whitelist entries include a length field, which specifies how many digits (0-6) the device will use to compare that entry to a card's IIN. The device ignores entries with lengths outside this range.

In addition, each entry in the PAN whitelist includes a flag that indicates whether the device will include the full PAN or a portion of the PAN when it sends the host **Notification 0x04::0x11 - MSR Card Data Available** or a response to **Command 0x05::0x01 - Request PAN**.

Table 4-49 - Message Structure for Command 0x03::0x80 - Get PAN / MSR Whitelist

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	80	**Command ID Data Object (Tag C2)** = 0x80 Read PAN / MSR Whitelist
C4	01	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Subcommand 0x80 = Read PAN Whitelist 0x81 = Read MSR Whitelist	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-50 - Response to Command 0x03::0x80 - Get PAN / MSR Whitelist

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	03	**Application ID Data Object (Tag C1)** = 0x03 Device Configuration Messages
C2	01	80	**Command ID Data Object (Tag C2)** = 0x80 Read PAN / MSR Whitelist
C3	01	**Result Code Data Object (Tag C3)** = 0x00 = OK / Done	

Tag	Len	Value(s) / Description
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Subcommand 0x80 = Read PAN Whitelist 0x81 = Read MSR Whitelist For subcommand 0x80, Bytes 1..64 contain the 8 entries in the PAN whitelist at 8 bytes per entry as follows: • 1 byte: Number of card IIN digits required for a match. • 6 bytes: IIN Compare Digits. • 1 byte: Flag for PAN availability. If set to "0" the device will make only the 12 digits required by an external EPP available to the host. If set to "1" the device will make the full PAN available. For subcommand 0x81, Bytes 1..56 contain the 8 entries in the MSR whitelist at 7 bytes per entry as follows: • 1 byte: Number of card IIN digits required for a match.. • 6 bytes: IIN Compare Digits.

4.7 Application Group 0x04 - Magnetic Stripe Reader (MSR) Messages

4.7.1 Notification 0x04::0x11 - MSR Card Data Available

When a cardholder swipes a magnetic stripe card, the device sends this notification to inform the host that card data is available. After receiving this notification, the host should call **Command 0x04::0x12 - Request MSR Card Data** to get the data.

Table 4-51 - Message Structure for Notification 0x04::0x11 - MSR Card Data Available

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	04	**Application ID Data Object (Tag C1)** = 0x04 MSR Messages
C2	01	11	**Command ID Data Object (Tag C2)** = 0x11 MSR Card Data Available
C4	02	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Card type 0x00 = Other 0x01 = ISO 0x02 = AAMVA Byte 1 Card whitelist comparison result 0x00 = The card is not in whitelist, encrypted data will be sent 0x01 = The card is in whitelist, unencrypted data will be sent Byte 2 PAN status 0x00 = PAN is not available 0x01 = PAN is in whitelist, and full PAN is available 0x02 = PAN is in whitelist, and only the 12 digits required for PIN block construction are available	

4.7.2 Command 0x04::0x12 - Request MSR Card Data

The host uses this command to request MSR data after receiving **Notification 0x04::0x11 - MSR Card Data Available**.

Table 4-52 - Message Structure for Command 0x04::0x12 - Request MSR Card Data

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	04	**Application ID Data Object (Tag C1)** = 0x04 MSR Messages
C2	01	12	**Command ID Data Object (Tag C2)** = 0x12 Request MSR Card Data

Table 4-53 - Response to Command 0x04::0x12 - Request MSR Card Data

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	04	**Application ID Data Object (Tag C1)** = 0x04 MSR Messages

Tag	Len	Value(s) / Description	
C2	01	12	**Command ID Data Object (Tag C2)** = 0x12 Request MSR Card Data
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = If Result Code is OK (00), **Data Object F4 - Magnetic Stripe Reader Card Data**, otherwise a 1-byte error code	

4.8 Application Group 0x05 - PAN Messages

4.8.1 Command 0x05::0x01 - Request PAN

The host uses this command to request PAN.

Table 4-54 - Message Structure for Command 0x05::0x01 - Request PAN

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	05	**Application ID Data Object (Tag C1)** = 0x05 PAN Messages
C2	01	01	**Command ID Data Object (Tag C2)** = 0x01 Request PAN

Table 4-55 - Response to Command 0x05::0x01 - Request PAN

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	05	**Application ID Data Object (Tag C1)** = 0x05 PAN Messages
C2	01	01	**Command ID Data Object (Tag C2)** = 0x01 Request PAN
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done
C4	Calculated	Byte 0 PAN Status 0x00 = PAN is not in whitelist, can not be transferred 0x01 = PAN is in whitelist, full PAN is available 0x02 = PAN is in whitelist, 12 digits of PAN are available for PIN block construction Byte 1 PAN Length Bytes 2..n PAN Value (if available)	

4.9 Application Group 0x07 - EMV L2 Contact Messages (Chip Card L2 Mode Only)

4.9.1 Command 0x07::0x00 - EMV L2 Start Transaction

The host uses this command to start an EMV L2 transaction.

Table 4-56 - Message Structure for Command 0x07::0x00 - EMV L2 Start Transaction

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	00	**Command ID Data Object (Tag C2)** = 0x00 EMV L2 Start Transaction
C4	19	**Data Field Data Object (Tag C4 or E0)** = Byte 0: Maximum Process Time Specifies the maximum time, in seconds, for user interaction events to complete while processing a transaction. Values from 0x01 to 0xFF (1 to 255 seconds) are allowed. The timer starts at the beginning of each event. If the cardholder does not take action within the specified time, the transaction will process as follows: • User card insertion timeout: The transaction terminates. • User language selection timeout: The transaction continues with the default language. • User application selection timeout: The transaction terminates. Byte 1 Card Type to Read: 0x01 = Magnetic Stripe (not supported at this time) 0x02 = Contact chip card 0x04 = Contactless chip card (not supported at this time) Magnetic Stripe and contact chip card can be enabled at the same time; however, if a cardholder swipes a magnetic stripe card in this mode, the device will send data in MagneSafe V5 format and the EMV transaction will be cancelled. Byte 2 Transaction Options: 0x00 = Normal 0x01 = Bypass PIN (not supported on this device) 0x02 = Force Online (not supported on this device) Bytes 3..8 Transaction Amount: EMV Tag 9F02, format n12, 6 bytes. If Byte 9 Transaction Type is set to Refund (0x20), the Transaction Amount must be zero. Byte 9 Transaction Type: 0x00 = Purchase (covers transaction types Payment, Goods, and Services) 0x02 or 0x09 = Cash back (0x09 only supported when using contactless) 0x20 = Refund (converts internally to type 0x00 Purchase, Bytes 3..8 are is ignored and set to $0.00) Bytes 10..15 Cash Back Amount: Cash back amount. If non-zero, use EMV Tag 9F03, format n12, 6 bytes. For Transaction Type Refund (0x20) this must be zero. Bytes 16..17 Transaction Currency Code (EMV Tag 5F2A, format n4, 2 bytes) Valid values:	

Tag	Len	Value(s) / Description
		0x0000 = Use Terminal Settings currency code 0x0840 = US Dollar 0x0978 = Euro Byte 18 Level of Transaction Status Notifications: Set the host desired level of transaction notifications: 0x00 = Termination status only (normal termination, card error, timeout, host cancel) 0x01 = Major status changes (terminations plus card insertions and waiting for user) 0x02 = All status changes (documents the entire transaction flow)

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-57 - Response to Command 0x07::0x00 - EMV L2 Start Transaction

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	00	**Command ID Data Object (Tag C2)** = 0x00 EMV L2 Start Transaction
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK

4.9.2 Command 0x07::0x02 - EMV L2 User Selection Result

The host uses this command to report the cardholder's or operator's selection in response to the device's **Notification 0x07::0x82 - EMV L2 User Selection Request**. In response to each possible selection, the device will behave according to EMV rules.

Table 4-58 - Message Structure for Command 0x07::0x02 - EMV L2 User Selection Result

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	02	**Command ID Data Object (Tag C2)** = 0x02 EMV L2 User Selection Result
C4	02	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Selection Status: 0x00 = User Selection Request completed, see Selection Result 0x01 = User Selection Request aborted, canceled by user 0x02 = User Selection Request aborted, timeout Byte 1 Selection Result: Contains the value of the menu item the cardholder or operator selected	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-59 - Response to Command 0x07::0x02 - EMV L2 User Selection Result

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	02	**Command ID Data Object (Tag C2)** = 0x02 EMV L2 User Selection Result
C3	01	**Result Code Data Object (Tag C3)** = 0x00 = Success, the Selection Result was received 0x01 = Invalid Selection Status 0x02 = Invalid Selection Result 0x03 = Failure, no transaction currently in progress	

4.9.3 Command 0x07::0x03 - EMV L2 Online Processing Result / Acquirer Response

The host uses this command to inform the device of the result of online processing. It will contain ARPC, Script 1, and Script 2 data.

Table 4-60 Message Structure for Command 0x07::0x03 - EMV L2 Online Processing Result / Acquirer Response

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	03	**Command ID Data Object (Tag C2)** = 0x03 EMV L2 Online Processing Result
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = See **Appendix C ARPC Response from Online Processing (EMV Only)**	

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-61 - Response to Command 0x07::0x03 - EMV L2 Online Processing Result / Acquirer Response

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	03	**Command ID Data Object (Tag C2)** = 0x03 EMV L2 Online Processing Result
C3	01	**Result Code Data Object (Tag C3) =** 0x00 OK / Done	

4.9.4 Command 0x07::0x04 - EMV L2 Cancel Transaction

The host uses this command to cancel an EMV transaction while the device is waiting for the cardholder to insert a card.

Table 4-62 - Message Structure for Command 0x07::0x04 - EMV L2 Cancel Transaction

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 EMV L2 Cancel Transaction

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-63 - Message Structure for Command 0x07::0x04 - EMV L2 Cancel Transaction

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 EMV L2 Cancel Transaction
C3	01	**Result Code Data Object (Tag C3) =** 0x00 = Success, the transaction was cancelled 0x8D = Failure, no transaction currently in progress 0x8F = Failure, transaction in progress, card already inserted	

4.9.5 Command 0x07::0x06 - EMV L2 Get Contact Terminal Configuration

The host uses this command to read EMV Contact Terminal configuration data.

Table 4-64 - Message Structure for Command 0x07::0x06 - EMV L2 Get Contact Terminal Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	06	**Command ID Data Object (Tag C2)** = 0x06 EMV L2 Get Contact Terminal Configuration
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0: Slot Number Must be 0x01 Byte 1: Operation 0x00 = Read Operation 0x0F = Read All Tags of selected slot Byte 2: Database Selector 0x00 = EMV Contact L2 0x01 = PayPass (for future release) 0x02 = payWave (for future release) 0x03 = Expresspay (for future release) 0x04 = D-PAS (for future release) Bytes 3..n: 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 4-65 - Response to Command 0x07::0x06 - EMV L2 Get Contact Terminal Configuration

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	06	**Command ID Data Object (Tag C2)** = 0x06 EMV L2 Get Contact Terminal Configuration
C3	01	**Result Code Data Object (Tag C3)** = 0x00 = Success, the read completed 0x93 = Failure, invalid slot number field 0x94 = Failure, invalid Operation field	

Tag	Len	Value(s) / Description
		0x95 = Failure, invalid Database Selector field 0x96 = Failure, invalid Tag to Read field
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0..1 Message Length Two byte hex, most significant byte first. This gives the total length of the EMV Terminal Configuration message that follows. Byte 2..N 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 will be: DFDF26, the Configuration Label DFDF47, the Database Checksum

4.9.6 Command 0x07::0x08 - EMV L2 Get Contact Application Configuration

The host uses this command to read back EMV contact application configuration data.

Table 4-66 - Message Structure for Command 0x07::0x08 - EMV L2 Get Contact Application Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	08	**Command ID Data Object (Tag C2)** = 0x08 EMV L2 Get Contact Application Configuration
C4	Calculated		**Data Field Data Object (Tag C4 or E0)** = Byte 0: Slot Number Must be 0x01 Byte 1: Operation 0x00 = Read Operation 0x0F = Read All Tags of selected slot Byte 2: Database Selector 0x00 = EMV Contact L2 0x01 = PayPass (for future release) 0x02 = payWave (for future release) 0x03 = Expresspay (for future release) 0x04 = D-PAS (for future release) Bytes 3..n: 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 4-67 - Response to Command 0x07::0x08 - EMV L2 Get Contact Application Configuration

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	08	**Command ID Data Object (Tag C2)** = 0x08 EMV L2 Get Contact Application Configuration

Tag	Len	Value(s) / Description
C3	01	**Result Code Data Object (Tag C3)** = 0x00 = Success, the read completed 0x93 = Failure, invalid slot number field 0x94 = Failure, invalid Operation field 0x95 = Failure, invalid Database Selector field 0x96 = Failure, invalid Tag to Read field
C4	Calculated	Byte 0..1: Message Length Two byte hex, most significant byte first. This gives the total length of the EMV Terminal Configuration message that follows. Byte 2..n 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 will be: DFDF26, the Configuration Label DFDF47, the Database Checksum

4.9.7 Command 0x07::0x0A - EMV L2 Get CA Public Key

The host uses this command to read EMV Certificate Authority Public Key data.

Table 4-68 - Message Structure for Command 0x07::0x0A - EMV L2 Get CA Public Key

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	0A	**Command ID Data Object (Tag C2)** = 0x0A EMV L2 Get CA Public Key
C4	Calculated		**Data Field Data Object (Tag C4 or E0)** = Byte 0: Slot Number Must be 0x01 Byte 1: Operation 0x00 = Read Operation 0x0F = Read All Tags of selected slot Byte 2: Database Selector 0x00 = EMV Contact L2 0x01 = PayPass (for future release) 0x02 = payWave (for future release) 0x03 = Expresspay (for future release) 0x04 = D-PAS (for future release) Bytes 3..n: 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 4-69 - Response to Command 0x07::0x0A - EMV L2 Get CA Public Key

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	0A	**Command ID Data Object (Tag C2)** = 0x0A EMV L2 Get CA Public Key
C3	01		**Result Code Data Object (Tag C3)** = 0x00 = Success, the read completed 0x93 = Failure, invalid slot number field 0x94 = Failure, invalid Operation field 0x95 = Failure, invalid Database Selector field 0x96 = Failure, invalid Tag to Read field

Tag	Len	Value(s) / Description
C4	Calculated	Byte 0..1: Message Length Two byte hex, most significant byte first. This gives the total length of the EMV Terminal Configuration message that follows. Byte 2..n 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 will be: DFDF26, the Configuration Label DFDF47, the Database Checksum

4.9.8 Command 0x07::0x11 - EMV L2 Read EMV Configuration

The host uses this command to read back EMV configuration data. Descriptions of the tags can be found in ***EMV Integrated Circuit Card Specifications for Payment Systems v4.3***.

Table 4-70 - Message Structure for Command 0x07::0x11 - EMV L2 Read EMV Configuration

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	11	**Command ID Data Object (Tag C2)** = 0x11 EMV L2 Read EMV Configuration
C4	Calculated		Byte 0: Database Selector 0x00 = EMV Contact L2 0x01 = PayPass (for future release) 0x02 = payWave (for future release) 0x03 = Expresspay (for future release) 0x04 = D-PAS (for future release)

Table 4-71 - Response to Command 0x07::0x11 - EMV L2 Read EMV Configuration

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	11	**Command ID Data Object (Tag C2)** = 0x06 EMV L2 Read EMV Configuration
C3	01	**Result Code Data Object (Tag C3)** = 0x00 = Success, the read completed 0x95 = Failure, invalid Database Selector field	
C4	01	Byte 0: Configuration Identifier, a one byte field that specifies one of the following configurations identified in the device's Implementation Conformance Statement (ICS). 0 = C0: Online Only with SDA, DDA and CDA enabled, tag 9F35 set to 21, tag 9F33 set to 20 28 C8 1 = C1: Online Only with SDA, DDA and CDA disabled, tag 9F35 set to 21, tag 9F33 set to 20 28 00 2 = C2: Offline/Online with SDA, DDA and CDA enabled, tag 9F35 set to 22, tag 9F33 set to 20 28 C8	

4.9.9 Command 0x07::0x80 - EMV L2 Transaction Status

The host uses this command to get the ongoing status of a transaction it has initiated using **Command 0x07::0x00 - EMV L2 Start Transaction**.

Table 4-72 - Message Structure for Command 0x07::0x80 - EMV L2 Transaction Status

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	80	**Command ID Data Object (Tag C2)** = 0x80 EMV L2 Transaction Status

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-73 - Response to Command 0x07::0x80 - EMV L2 Transaction Status

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	80	**Command ID Data Object (Tag C2)** = 0x80 EMV L2 Transaction Status
C3	01	**Result Code Data Object (Tag C3)** = 0x00 OK / Done	
C4	05	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Indicates the event that triggered this notification: 0x00 = No events since start of transaction 0x01 = Card Inserted 0x02 = Card Error 0x03 = Transaction Progress Change 0x04 = Waiting for User Response 0x05 = Timed Out 0x06 = Transaction Terminated 0x07 = Host Cancelled Transaction 0x08 = Card Removed Byte 1 indicates the remaining time available, in seconds, for the indicated operation to complete. The timeout is set by the host when calling **Command 0x07::0x00 - EMV L2 Start Transaction** Byte 2 indicates the current processing stage for the transaction: 0x00 = No transaction in progress 0x01 = Waiting for cardholder to insert card 0x02 = Powering up the card 0x03 = Selecting the application 0x04 = Waiting for user language selection 0x05 = Waiting for user application selection 0x06 = Initiating application 0x07 = Reading application data 0x08 = Offline data authentication	

Tag	Len	Value(s) / Description
		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 online processing response 0x11 = Transaction complete 0x12 = Transaction error 0x13 = Transaction approved 0x14 = Transaction declined 0x15 = Transaction cancelled by MSR swipe 0x16 = EMV error - Conditions Not Satisfied 0x17 = EMV error - Card Blocked 0x18 = Application selection failed 0x19 = EMV error - Card Not Accepted 0x1A = Empty Candidate List 0x1B = Application Blocked Bytes 3..4 are reserved

4.9.10 Notification 0x07::0x81 - EMV L2 Display Message Request

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

Table 4-74 - Message Structure for Notification 0x07::0x81 - EMV L2 Display Message Request

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	81	**Command ID Data Object (Tag C2)** = 0x81 Display Message Request
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = A string of data up to 1024 bytes containing message to be displayed on the host	

4.9.11 Notification 0x07::0x82 - EMV L2 User Selection Request

The device uses this notification to inform the host that a user 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 **Command 0x07::0x02 - EMV L2 User Selection Result** to send the selection result and inform the device that the transaction can proceed.

Table 4-75 - Message Structure for Notification 0x07::0x82 - EMV L2 User Selection Request

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	82	**Command ID Data Object (Tag C2)** = 0x82 EMV L2 User Selection Request
C4	03	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Selection Type specifies what kind of selection request this is: 0x00 = Payment Brand Application Selection 0x01 = Language Selection Byte 1 Timeout specifies the maximum time, in seconds, allowed to complete the selection process. If this time is exceeded, the host should send **Command 0x07::0x02 - EMV L2 User Selection Result** with the Selection Status field set to 0x02 (User Selection Request aborted, timeout), after which the transaction will be aborted and an appropriate Transaction Status will be available. Value 0 (User Selection Request completed) is not allowed in this case. Byte 2 Menu Items is a variable length 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 will display the menu items to the cardholder, then, after the cardholder makes a selection, call **Command 0x07::0x02 - EMV L2 User 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 list title only.	

4.9.12 Notification 0x07::0x83 - EMV L2 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 **Command 0x07::0x03 - EMV L2 Online Processing Result / Acquirer Response** to inform the device it can proceed with the transaction.

Table 4-76 - Message Structure for Notification 0x07::0x83 - EMV L2 ARQC Message

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	83	**Command ID Data Object (Tag C2)** = 0x83 ARQC Message
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Bytes 0..n ARQC Message	

The ARQC Message 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<len> /*container for ARQC */
                     DFDF53<len><value> /*fallback indicator */
                     5F20<len><value> /*cardholder name */
                     5F30<len><value> /*service code */
                     DFDF4D<len><value> /* Mask T2 ICC Data */
                     DFDF52<len><value> /* card type */
                     F8<len> /*container tag for encryption */
                          DFDF59(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, always set to zeroes)
```

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

```
           FC<len>/* container for encrypted generic data */
                <tags defined by DFDF02 >
                F5<len>/* container tag for encrypted PIN data */
                          99(Encrypted PIN DATA)<len><val>
                          DFDF41(PIN KSN Data)<len><val>
                          DFDF42(PIN EncryptionType)<len><val>
                And more…
```

4.9.13 Notification 0x07::0x84 - EMV L2 Transaction Result

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

Table 4-77 - Message Structure for Notification 0x07::0x84 - EMV L2 Transaction Result

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	84	**Command ID Data Object (Tag C2)** = 0x84 EMV L2 Transaction Result
C4	Calculated	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Signature Required. This field indicates whether a cardholder signature is required to complete the transaction: 0x00 = No signature required 0x01 = Signature required. If a signature is required, the host should acquire the signature from the cardholder as part of the transaction data. Bytes 1..2 Batch Data Length. Two byte binary, most significant byte first. This gives the total length of the Batch Data message that follows, excluding padding and CBC-MAC. Byte 3 Batch Data: See **Appendix D Transaction Result Message - Batch Data Format (EMV Only)**. It is expected that the host will save this data as a record of the transaction.	

4.9.14 Notification 0x07::0x87 - EMV L2 PIN Entry Show Prompt Request

The device uses this notification to request that the host display a PIN entry related prompt on the host's display.

Table 4-78 - Message Structure for Notification 0x07::0x87 - EMV L2 PIN Entry Show Prompt Request

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	87	**Command ID Data Object (Tag C2)** = 0x87 EMV L2 PIN Entry Show Prompt Request
C4	01	**Data Field Data Object (Tag C4 or E0)** = Byte 0 Prompt ID specifies which prompt the device want the secure host to display: 0x01 = AMOUNT 0x02 = AMOUNT OK? 0x03 = APPROVED 0x04 = CALL YOUR BANK 0x05 = CANCEL OR ENTER 0x06 = CARD ERROR 0x07 = DECLINED 0x08 = ENTER AMOUNT 0x09 = ENTER PIN 0x0A = INCORRECT PIN 0x0B = INSERT CARD 0x0C = NOT ACCEPTED 0x0D = PIN OK 0x0E = PLEASE WAIT 0x0F = PROCESSING ERROR 0x10 = REMOVE CARD 0x11 = USE CHIP READER 0x12 = USE MAG STRIPE 0x13 = TRY AGAIN	

4.9.15 Notification 0x07::0x88 - EMV L2 PIN CVM Request

The device uses this notification to request a PIN block from the host when no encrypting PIN pad (EPP) is attached to the device.

Table 4-79 - Message Structure for Notification 0x07::0x88 - EMV L2 PIN CVM Request

Tag	Len	Value(s) / Description	
C0	01	03	**Message Type Data Object (Tag C0)** = 0x03 Notification
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	88	**Command ID Data Object (Tag C2)** = 0x88 PIN CVM Request Notification

Table 4-80 - Response to Notification 0x07::0x88 - EMV L2 PIN CVM Request

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	07	**Application ID Data Object (Tag C1)** = 0x07 EMV L2 Contact Messages
C2	01	88	**Command ID Data Object (Tag C2)** = 0x88 PIN CVM Request Notification
C3	01	**Result Code Data Object (Tag C3) =** 0x00 = OK / Done	
C4	Calculated	Byte 0 Status 0x00 = Operation successful 0x01 = General failure 0x02 = User cancelled operation. 0x03 = Operation timed out 0x04 = CVM failed Bytes 1..n PIN block if available.	

4.10 Application Group 0x08 - Manufacturer Configuration Messages

4.10.1 Command 0x08::0x03 - Re-Activate Device

The host uses this command to set the device's state to **installed**. The device will only accept this command if it is in the **removed** state. Use **Command 0x01::0x04 - Get Device Status** to read the state.

Table 4-81 - Message Structure for Command 0x08::0x03 – Re-activation command

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	08	**Application ID Data Object (Tag C1)Application ID Data Object (Tag C1)** = 0x08 Manufacturer Configuration Message
C2	01	03	**Command ID Data Object (Tag C2)** = 0x03 Re-Activate Device

If an error occurs, the device will terminate the command and report the error using an **ACK Response** containing the result code. For a full list of error codes, see **2.1.4 Result Code Data Object (Tag C3)**. If no error occurs, the device will respond as follows:

Table 4-82 - Response to Command 0x08::0x03 – Re-activation command

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	08	**Application ID Data Object (Tag C1)Application ID Data Object (Tag C1)** = 0x08 Manufacturer Configuration Message
C2	01	03	**Command ID Data Object (Tag C2)** = 0x03 Re-Activate Device
C3	01	00	**Result Code Data Object (Tag C3) =** 0x00 OK / Done

4.10.2 Command 0x08::0x04 - Read Dismount Switch State

The host uses this command to read current state of the device's dismount switch, and can be used in any device dismount status. This command is useful for checking the device's mounting condition to make sure it is mounted properly before activating it.

Table 4-83 - Message Structure for Command 0x08::0x04 – Read dismount switch state command

Tag	Len	Value(s) / Description	
C0	01	01	**Message Type Data Object (Tag C0)** = 0x01 Command
C1	01	08	**Application ID Data Object (Tag C1)Application ID Data Object (Tag C1)** = 0x08 Manufacturer Configuration Message
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 Read Dismount Switch State

If the dismount switch is mounted properly, then the device will respond with 0x00. Otherwise, the device will respond with 0x01 for representing that the switch is not mounted. In case of receiving 0x01, please check mounting condition for proceeding activation process. For a full list of result codes, see **2.1.4 Result Code Data Object (Tag C3)**. The device will respond as follows:

Table 4-84 - Response to Command 0x08::0x04 – Read dismount switch state command

Tag	Len	Value(s) / Description	
C0	01	02	**Message Type Data Object (Tag C0)** = 0x02 Response
C1	01	08	**Application ID Data Object (Tag C1)Application ID Data Object (Tag C1)** = 0x08 Manufacturer Configuration Message
C2	01	04	**Command ID Data Object (Tag C2)** = 0x04 Read Dismount Switch State
C3	01	00 or 01	**Result Code Data Object (Tag C3) =** 0x00 = Device registers as mounted 0x01 = Device registers as not mounted

Appendix A Examples

Reserved

Appendix B MagTek Custom EMV Tags (EMV Only)

In addition to the standard EMV tags documented in ***EMV 4.3, Book 3, Annex A***, MagTek provides additional custom tags with the device. These are used with **Command 0xA1 - Access EMV Tags**, **Command 0xA2 - Request Start EMV Transaction**, and **Command 0xAB - Request EMV Transaction Data (MAC-MSR)**. The custom tags are listed in **Table 4-85**. The characters used in the "Format" column are described in ***EMV 4.3, Book 4, Section 4.3***.

Table 4-85 - MagTek Custom EMV Tags

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

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

Tag	Description	Default (HEX)	Format	Length
DFDF24	Requested Transaction Type	*	b	1
DFDF25	Unique and permanent serial number assigned to the IFD by the manufacturer	USIP SN	b	8
DFDF26	Device & EMV Application Database Label		b	16
DFDF27	Device & EMV Application Database Checksum	Read-only, calculated by device	b	20
DFDF28	CAPK Database Label		b	16
DFDF29	CAPK Database Checksum	Read-only, calculated by device	b	20
DFDF2D	Supported Terminal Languages	65 6e 66 72 69 74 64 65 65 73	b	10
DFDF30	Masked T1 Status	*	b	1
DFDF31	Masked T1	*	a	var
DFDF32	Masked T2 Status	*	b	1
DFDF33	Masked T2	*	a	var
DFDF34	Masked T3 Status	*	b	1
DFDF35	Masked T3	*	a	var
DFDF36	Encrypted T1 Status	*	b	1
DFDF37	Encrypted T1	*	b	var
DFDF38	Encrypted T2 Status	*	b	1
DFDF39	Encrypted T2	*	b	var
DFDF3A	Encrypted T3 Status	*	b	1
DFDF3B	Encrypted T3	*	b	var
DFDF3C	Encrypted MagnePrint	*	b	56
DFDF3D	MagneSafe 2.0 Status	*	b	8
DFDF3F	CAPK Tag	*	b	var
DFDF40	Signature Required: 0x01 = Signature Required 0x80 = CBC-MAC checked in ARQC online response	*	b	1
DFDF41	PIN KSN	*	b	10
DFDF42	PIN Encryption Type: 0xxx xxxx = Fixed key 1xxx xxxx = DUKPT key xx00 xxxx = TDES	*	b	1

Tag	Description	Default (HEX)	Format	Length
	xx01 xxxx = AES xxxx xx00 = Data variant xxxx xx01 = PIN variant xxxx xx10 = MAC variant			
DFDF43	MagnePrint Status Data	*	b	4
DFDF44	Encrypted PAN Data	*	b	var
DFDF4D	Masked ICC Track2 data	*	a	var
DFDF50	MSR KSN	*	b	10
DFDF51	MSR Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF52	Card Type Reported as clear text in ARQC with possible values 01 or 07.	-	b	1
DFDF53	Fallback Indication 0x00=No fallback or missing tag 0x81=MSR Fallback used 0x01=Technical Fallback used	*	b	1
DFDF54	MAC KSN	*	b	10
DFDF55	MAC Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF56	Encrypted Transaction Data KSN	*	b	10
DFDF57	Encrypted Transaction Data Encryption Type (see DFDF42 for bit definitions)	*	b	1
DFDF58	Number of Bytes of Padding in F8	*	b	1
DFDF59	Encrypted Data Primitive	*	b	var
DFDF61	BIN Table Slot 1	00 00 00 00 00 00	b	6
DFDF62	BIN Table Slot 2	00 00 00 00 00 00	b	6
DFDF63	BIN Table Slot 3	00 00 00 00 00 00	b	6
DFDF64	BIN Table Slot 4	00 00 00 00 00 00	b	6
DFDF65	BIN Table Slot 5	00 00 00 00 00 00	b	6
DFDF66	BIN Table Slot 6	00 00 00 00 00 00	b	6
DFDF67	Acquirer Terminal Config - Fallback (0=Fallback Not Supported, 1=Fallback Supported)	01	b	1
DFDF68	Acquirer Terminal Config - PIN Bypass (0=PIN Bypass Not Supported, 1-PIN Bypass Supported)	01	b	1
DFDF70	Terminal Action Code - Default	00 00 00 00 00 00	b	5

Tag	Description	Default (HEX)	Format	Length
DFDF71	Terminal Action Code - Denial	00 00 00 00 00 00	b	5
DFDF72	Terminal Action Code - Online	00 40 00 00 00 00	b	5
DFDF73	Payment Brand Account Type 0x00 = Default 0x01 = Credit, Debit, or Default 0x02 = Debit 0x03 = Credit	00	b	1

* - Value is based on the ongoing transaction

Appendix C ARPC Response from Online Processing (EMV Only)

This section gives the format of the data for **Command 0x07::0x03 - EMV L2 Online Processing Result**. The host sends this command to the device in response to **Notification 0x07::0x83 - EMV L2 ARQC Message**. The data is a TLV 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)
```

Appendix D Transaction Result Message - Batch Data Format (EMV Only)

This section gives the format of the data the device uses for **Notification 0x07::0x84 - EMV L2 Transaction Result**. The contents of tag DFDF1A will contain one of the following transaction statuses:

- 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
- 0xFF = Unknown

When the device is set to not encrypt, the 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 */
                  F2<len>/* container for Batch Data */
                        … /* Batch 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 */
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

When the device is set to encrypt, the TLV data object contains 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 */
                  F8<len> /* container tag for encryption */
                        DFDF59(Encrypted Data Primative)<len><Encrypted
Data val (Decrypt data to read tags)>
                        DFDF56(Encrypted Transaction Data KSN)<len><val>
```

```
                    DFDF57(Encrypted Transaction Data Encryption
Type)<val>
                    DFDF58(# of bytes of padding in DFDF59)<len><val>
               F7<len>/* container for Merchant Data */
                    … /* < Merchant Data tags */
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

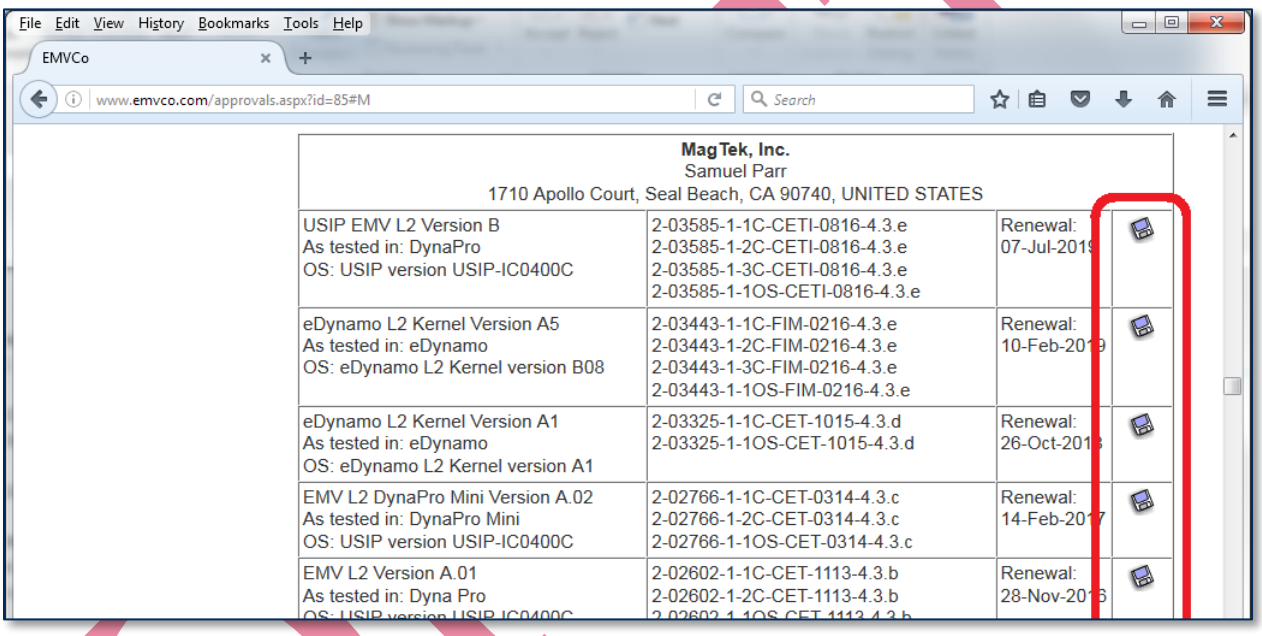
The value inside tag DFDF59 is encrypted and contains the following after decryption:

```
          FC<len>/* container for encrypted generic data */
               F2<len>/* container for Batch Data */
                    … /* Batch Data tags (defined by DFDF17) */
               F3<len>/* container for Reversal Data, if any */
                    … /* Reversal Data tags (defined by DFDF05)*/
```

Appendix E EMV Configurations (EMV Only)

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

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



Appendix F Factory Defaults

This section lists device configuration tags available in the device for use with the commands in **Application Group 0x07 - EMV L2 Contact Messages (Chip Card L2 Mode Only)**.

F.1 Certificate Authority Public Keys (EMV Only)

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

F.2 EMV Contact Factory Defaults (EMV Only)

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

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

F.2.1 EMV Contact Terminal Factory Defaults

Table 4-86 - EMV Contact Terminal Factory Defaults

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

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

F.2.2 EMV Contact Payment Brand Factory Defaults

Table 4-87 - EMV Contact Payment Brand Factory Defaults - Slot 0

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

Slots 1 through 9 are left empty, except for devices that support **multiple payment brand defaults**, where application slots 1 through 9 are set as follows:

Table 4-88 - EMV Contact Payment Brand Factory Defaults - Slot 1

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 03 10 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 03 10 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00

Tag Description	Tag	Configurable	Default Value(hex)
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-89 - EMV Contact Payment Brand Factory Defaults - Slot 2

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 03 20 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 03 20 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-90 - EMV Contact Payment Brand Factory Defaults-Slot 3

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 04 20 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 04 20 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00

Tag Description	Tag	Configurable	Default Value(hex)
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-91 - EMV Contact Payment Brand Factory Defaults-Slot 4

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 03 30 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 03 30 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-92 - EMV Contact Payment Brand Factory Defaults-Slot 5

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 65 10 10
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 65 10 10
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01

Tag Description	Tag	Configurable	Default Value(hex)
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-93 - EMV Contact Payment Brand Factory Defaults-Slot 6

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 25 01 05 01
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 25 01 05 01
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-94 - EMV Contact Payment Brand Factory Defaults-Slot 7

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 25 01 05 08
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 25 01 05 08
Floor Limit	0x9F1B	MagTek	00 00 27 10

Tag Description	Tag	Configurable	Default Value(hex)
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-95 - EMV Contact Payment Brand Factory Defaults-Slot 8

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 25 01 05 09
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01
Application AID	0x9F06	MagTek	A0 00 00 00 25 01 05 09
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Table 4-96 - EMV Contact Payment Brand Factory Defaults-Slot 9

Tag Description	Tag	Configurable	Default Value(hex)
Application DF Name	0x84	MagTek	A0 00 00 00 25 01 05 0A
TDOL	0x97	MagTek	5F2403
Acquirer ID	0x9F01	MagTek	00 00 00 00 00 01

Tag Description	Tag	Configurable	Default Value(hex)
Application AID	0x9F06	MagTek	A0 00 00 00 25 01 05 0A
Floor Limit	0x9F1B	MagTek	00 00 27 10
DDOL	0x9F49	MagTek	9F 37 04 5A 08 5F 34 01 9A 03
ASI (Application Select Indicator)	0xDFDF23	MagTek	01
Application Version	0x9F09	MagTek	00 8C
TAC - Default	0xDFDF70	MagTek	00 00 00 00 00
TAC - Denial	0xDFDF71	MagTek	00 00 00 00 00
TAC - Online	0xDFDF72	MagTek	00 40 00 00 00
Payment Brand Account Type	0xDFDF73	MagTek	03
Terminal Threshold Value	0xDFDF10	MagTek	00 00 00 00 40 00
Terminal Target Percentage	0xDFDF11	MagTek	32
Terminal Max Target Percentage	0xDFDF12	MagTek	46

Appendix G Language and Country Codes (EMV Only)

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

G.1 Terminal Country Codes

Table 4-97 - Terminal Country Codes

0840	United States
0250	France
0380	Italy
0724	Spain
0276	Germany

G.2 Terminal Language Codes

Table 4-98 - Terminal Language Codes

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