MAGTEK®
SECURITY FROM THE INSIDE

mDynamo

OEM Secure Card Reader Authenticator
Programmer's Manual (COMMANDS)

April 2024

Document Number:
D998200151-200

REGISTERED TO ISO 9001:2015

Table 0-1 - Revisions

Rev Number	Date	Notes
11	Oct 28, 2016	Initial release derived from D100003048
20	Mar 03, 2017	Derived from D100003048-13: Add fixed key features; improve clarity of encryption-related features; add command security information for mDynamo; replace cross-references to unpublished ***Audio Reader Communication Protocol_v1*** with in-place content; change "manual entry" to "keypad entry" for clarity; add missing example tables for extended commands in section **7.4**; misc. clarifications
21	Jul 25, 2017	Fix typo
22	Jun 11, 2020	Derived from D100003048-18: Add detail to **Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)**; fix section **5** to accommodate EMV-only devices; expose **Property 0x67 - EMV Data Encryption Variant (EMV Only)**; move KSN interpretation info to **Command 0x09 - Get Current TDES DUKPT KSN** to provide details for devices that do not have EMV; add Dynasty, kDynamo, mDynamo Contactless Module, pDynamo, tDynamo; remove vestigial Properties Per Device table from section **8** (now covered by section heading tags); Add **Property 0x52 - Host Poll Timeout (HID Only | KB Only)**; Add contactless default settings to **Appendix C EMV Terminal and Application Settings (EMV Only)**; Add cDynamo, Dynamag Duo; Add **Property 0x6D - EMV Contact Notification Configuration (Contact Only)**; Modify **Property 0x33 - Card Inserted (MSR Insert Only | Contact Only)**; Add **Command 0x51 - External LED Control (External LED Control Only)**; Change “insert a card” “swipe a card” and “tap a card” to “present payment” (reviewed by EL and HM); Tag relevant features and values as Only Contact / Online; Retrofit Unattended Operation feature, notably in **Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)**; Update device defaults in **Property 0x10 - Interface Type**; Add tDynamo, kDynamo, mDynamo Contactless; Major rewrite to streamline and add to **Table 1-2**; Add options in **Extended Command 0x030B - Read EMV Kernel Information** to retrieve contactless kernel version, checksum, and configuration info; Add **Extended Command 0x0305 - Modify Terminal Configuration (MAC)**; Add **Appendix C EMV Terminal and Application Settings (EMV Only)**; Correct example in **Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)**; Retrofit filtering out of properties by SureSwipe feature; Update contactless statuses in **Notification 0x0300 - Transaction Status / Progress Information**; Large update to **Table 1-2 - Device Features**; Update examples in **Extended Command**

		0x0305 - Modify Terminal Configuration (MAC), **Extended Command 0x0307 - Modify Application Configuration (MAC)**; Add Quick Chip option in **Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)**; Add tag 9F40 in **Extended Command 0x0311 - Read EMV Configuration (Contact Only)**; Update available configurations in **Extended Command 0x0310 - Modify EMV Configuration (MAC,** Contact Only), **Extended Command 0x0311 - Read EMV Configuration (Contact Only)**; Extract tags common to all EMV databases into **C.1 EMV Common Settings**; Major clarity rewrite of introduction to EMV transactions in section **7.4**; Remove legacy "original format" for EMV messages and merge EMV message formats under a single **Appendix B**; Separate iAP1 from iAP2 connection type; **Table 1-2** add tDynamo Pairing Modes and Custom Advertising; **Table 1-2** add Set Mask Service Code to P-series, I-65 w/V5, DynaWave; Add Multi-Language support for kDynamo; Create new common command and response documentation section **7.1**, remove from connection type / format sections; Stop requiring MAC for **Extended Command 0x030C - Set Date and Time (MAC)**; Add notification codes for MSR in **Notification 0x0300 - Transaction Status / Progress Information**; Section **B.1** add information about DFDF4D contents; Throughout, clarify ANS specification and key names/variants; Add clarifying cross-references between sections about masked and encrypted card data; Section **6.2.1** add code 0x29; Section **1.5** update JIS Capable feature for SPI Encrypting IntelliHead V5 and UART Enc IntelliHead V5; Add JIS Capable and Set Mask Service Code features to Dynamag and USB Enc IntelliHead V5; Add Quick Chip feature and more supporting information; Remove deprecated products including Flash reader, Home Banking (Dynamo LCD), iDynamo throughout; Remove deprecated connection types Proprietary Wireless and 30-pin throughout; Remove deprecated features Store and Forward, Custom Messages (redundant with Display),.and strip out properties no longer needed; Change feature Unattended Mode to OEM Features; Rotate feature table for space constraints; For clarity, remove redundant feature tags from section headings that implicitly inherit from parent sections; Section **4.3** add "C" as a security option; Throughout, clarify **Security Level 2** behavior and impact on MagnePrint values and security; Add Contactless Quick Chip feature and supporting detail, split EMV Quick Chip feature into Contact vs. Contactless; Add OEM Features feature and supporting information; Update **Table 1-2** and add iDynamo 5 (Gen II), iDynamo 6 and Power Management scheme PM7; Add External PIN Accessory Support feature; Add Application Selection Options feature and supporting detail; Add Dual

		USB Ports as an explicit feature to cover more devices; In **Notification 0x0300 - Transaction Status / Progress Information** add Progress Indicator 0x91 and others, tweak descriptions for clearer use in major clarity and completeness updates in section **7.4.2 About EMV L2 Transaction Flows (EMV Only)**; Add QuickPass Support feature and supporting information; Remove Dynasty; Refactor section **2.1.3** to improve severability between input report info for MSR vs. Notifications; Move terminal country codes and terminal language codes into tag tables instead of a dedicated appendix; Throughout, clarify Currency Codes are driven by ISO standard; Misc. clarifications and corrections.
200	April 02, 2024	Removed all instances of Tag **DFDF67**; Updated **Table 7-40** to remove MSR Falback; Update **Table 1-1** and **Table 1-2** to add mDynamo Gen II and RS-232 content, Add **2.2 How to Use UART and RS-232 Connections (RS-232 Only | UART Only), add section 3.2 How to Use SLIP Format (SLIP Only)**

Table of Contents

1 Introduction

1.1 About This Document

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

1.2 About SDKs

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

The SDKs and corresponding documentation include:

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

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

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

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

1.3 About Terminology

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

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

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

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

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

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

1.4 About Connections and Data Formats

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

Table 1-1 - Device Connection Types / Data Formats

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

Product / Connection	Audio	Bluetooth LE GATT	Bluetooth LE GATT KB	Bluetooth	Lightning iAP1	Lightning iAP2	RS-232 / UART	SPI	USB HID	USB KB
iDynamo 6						SLIP			HID	
kDynamo						SLIP				
mDynamo Gen I									HID	
mDynamo Gen II							SLIP			
P-series and I-65 w/V5									HID	Streaming
pDynamo		GATT							HID	
sDynamo						Streaming				
SPI Enc IntelliHead V5								Streaming		
tDynamo		GATT							HID	
UART Enc IntelliHead V5							Streaming			
uDynamo	TLV								HID	

1.5 About Device Features

The information in this document applies to multiple devices. When developing solutions that use a specific device or set of devices, integrators must be aware of each device's connection types, data formats, features, and configuration options, which affect the availability and behavior of some commands. **Table 1-2** provides a list of device features that may impact command availability and behavior. All section headings in this document include tags that indicate which features they apply to.

Table 1-2 - Device Features

Feature / Product	BulleT KB BulleT SPP	cDynamo	Dynamag, USB Enc IntelliHead V5	Dynamag Duo	DynaMAX	DynaPAD	DynaWave	eDynamo	iDynamo 5	iDynamo 5 (Gen II)	iDynamo 6	kDynamo	mDynamoGen I	mDynamo Gen II	P-series, I-65 w/V5	pDynamo	sDynamo	SPI Encrypting IntelliHead V5	tDynamo	UART Enc IntelliHead V5	uDynamo
MSR Swipe	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y	Y
MSR Insert	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
MSR 3 Tracks	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	
MSR Disable	N	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N
MSR Swap Tracks 1/3	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
MSR Embedded V5 Head	N	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	N	Y	N	Y	N	N
MSR Configurable MSR Variants		Y	Y	Y	Y		N	Y	Y	Y	Y	Y	Y	N		Y	Y		Y		
MSR Configurable MP Variants		N	N	N	Y		N	Y	N	N	Y	Y	Y	Y		Y	N		Y		
MSR SureSwipe		N	Y	Y	Y	Y	N	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N
MSR JIS Capable		Y	Y[3]	N	N	N	N	N	Y	N	N	N	N	N	N	N	Y	Y	N	Y	
MSR SHA-1		N	Y	Y	Y	Y	N	Y	N	N	N	N	N	N		Y	N		N		
MSR SHA-256		N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N		N		

MSR Configurable SHA		N	N	N	Y		N	Y	N	N	N	N	N	N			N		N		
MSR MagneSafe 2.0							N	Y		N	N	N	N	N		N			N		
Configurable Encryption Algorithm	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N		N	N	N
Set Mask Service Code	N	N	Y[2]	N	N	N	Y	N	N	N	N	N	N	N	Y[2]	N	N	Y[2]	N	N	N
Never Mask Service Code			N[2]				N	Y	Y	Y	Y	Y	Y	Y	N[2]		Y	N[2]	Y		
EMV Contact	N	N	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	N	N	N	N	Y	N	N
EMV Contactless	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N
EMV Offline ODA	N	N	N	N	N	N	Y	Y	N	N	N	N	Y	Y	N	N	N	N	N	N	N
EMV MSR Flow	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N
EMV Contact Quick Chip	N	N	N	N	N	N	N	Y[4]	N	N	Y	Y	Y[4]	Y[4]	N	N		N	Y	N	N
EMV Contactless Quick Chip	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N
QuickPass Support	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Application Selection Options	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N
External PIN Accessory Support	N	N	N	N	N	N	Y	N	N	N	Y	N	N	N	N	N	N	N	N	N	N
Keypad Entry	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Fixed Key	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N
MSR Secondary DUKPT Key	N	N	N	N	Y	N	Y	Y	N	N	N	N	Y	Y	N	Y	N	N	N	N	Y
Power Mgt Scheme (PM#)	1	N	N	N	2	N	N	3	N	N	7	5	N	N	N	6	N	N	5	N	4

Battery-Backed RTC							N	Y		N	N	N	N	N					N		
OEM Features	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Transaction Validation	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N
Display	N	N	N	N	N	Y*	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N
Multi-Language	N	N	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	N	N	N	Y	N	N
Tamper	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
Extended Commands	N	N	N	N	N	N	Y	Y	N	N	Y	Y	Y	Y	N	N	N	N	Y	N	N
Extended Notifications	N	N	N	N	N	N	Y	Y	N	N	Y	Y	Y	Y	N	N	N	N	Y	N	N
Dual USB Ports	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	Y	N	N
Pairing Modes	N	N	N	N	N	N	N	Y[5]	N	N	N	N	N	N	N	Y	N	N	Y	N	N
Custom Advertising	N	N	N	N	N	N	N	Y[6]	N	N	N	N	N	N	N	Y	N	N	Y	N	N
Configurable Lightning FID	N	Y	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	N
Auxiliary Ports	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N
Configurable Pushbutton	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	Y	N	N
External LED Control	N	N	N	N	N	N	N	N	N	N	N	N	Y[1]	Y [1]	N	N	N	N	N	N	N
Encrypt Bulk Data (b)	120	120	24	24	24	N	N	24	120	N	N	N	24	24	N	N	N	120	N	12	24

	1) This feature is available in mDynamo firmware revision 1000003358D00 (released August 2017) and newer. 2) This feature was introduced in SPI Encrypting IntelliHead V5 in firmware version 21042876C01 released July 2017, P-series and I-65 w/V5 in firmware version 21165822E01 released March 2018, Dynamag and USB Encrypting IntelliHead V5 in firmware version 21042840K00 released January 2019. 3) This feature is available in Dynamag and USB Enc IntelliHead V5 firmware version 21042840K00 (released January 2019) and newer. 4) EMV Contact Quick Chip is available in mDynamo firmware revision 1000003358F01 (released December 2017), eDynamo firmware revision 1000003354F00 (released October 2018), and newer. 5) Pairing Modes feature is available in eDynamo firmware revision 1000002650B01 and newer, except Bluetooth LE Property 0x13 which was added in 1000002650C01. 6) Custom Advertising feature is available in eDynamo firmware revision 1000002650C02 and newer, except Bluetooth LE Property 0x08 Configuration Bits "Never Advertise" and "USB Power Not Exit Airplane Mode" which were added in 1000002650C01.

2 Connection Types

Table 1-1 on page **12** includes a list of connection types available for each device. The following subsections provide details developers will need to communicate with the device using each connection type.

2.1 How to Use USB Connections (USB Only)

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

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

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

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

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

- tDynamo reports **0x001C**.
- kDynamo reports **0x001D**.
- DynaWave reports **0x001E**.
- MSR Swipe devices report PID **0x0011** when in HID mode.
- MSR Insert devices report PID **0x0013** when in HID mode.
- Audio devices report PID **0x0017** when in HID mode.
- Devices that implement a combination of EMV Contact / EMV Contactless / MSR swipe report PID **0x0019** when in HID mode.
- EMV-only devices (such as mDynamo and DynaWave) report PID **0x001A** when in HID mode.
- All devices report PID **0x0001** when in KB mode.
- Wireless USB device dongles report PID **0x0011** when in HID mode.
- Wireless USB device dongles report PID **0x0001** when in KB mode.
- Wireless USB devices report PID **0x0014** when plugged directly into the host with a USB cable.

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

All USB HID devices send and receive data using **Reports**. Each report can contain several sections, called **Usages**, each of which has its own unique four-byte (32-bit) identifier. The two most significant bytes of a usage are called the **usage page**, and the two least significant bytes are called the **usage ID**. Vendor-defined HID usages must have a usage page in the range **0xFF00 - 0xFFFF**, and it is common practice for related usage IDs share the same usage page. For these reasons, all usages for MagneSafe V5 devices use vendor-defined usage page **0xFF00**, **Magnetic Stripe Reader**.

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

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

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

2.1.2 How to Send Commands On the USB Connection

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

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

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

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

1) Choose the command to invoke from section **7 Commands**. Every command has a corresponding **Command Number** listed in the header of its documentation section.
2) Construct a **Command Request Data** value using the Request table in the documentation for the command.
3) Determine the length of the **Command Request Data** value, referred to as the **Command Request Data Length**.
4) Examine the device's Report Descriptor to determine what payload length the device expects for a **Set Feature Report** and **Get Feature Report** (the operating system libraries may refer to this length as the "Report Length" or "Report Count").
5) Pad the **Command Request Data** value with 0x00 so the total length of the payload is consistent with the Set Feature Report's Report Length / Report Count.
6) Construct a Set Report Structure using **Table 7-1** in section **7.1 About Commands**.
7) Send a **Set Feature Report** containing the finalized padded Set Report Structure. The call to send the report may succeed, fail on a timeout, or fail for some other reason.
8) If the call succeeds, send a **Get Feature Report** to retrieve the device's response in the Get Report Structure shown in **Table 7-2** in section **7.1 About Commands**.
9) Parse the Get Report Structure, and truncate the **Command Response Data** field to the provided **Command Response Data Length**.
10) Examine the **Result Code**, which is a one-byte value the device sends to indicate success or the failure mode of the command. See section **7.2 About Result Codes** for more detail.
11) Parse the truncated **Command Response Data** field using the Response table in the documentation for the command.

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

(Extended Commands Only)

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

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

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

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

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

- **Notification Messages Sent from Device to Host (Extended Notifications Only)** use **Input Report ID 2** for asynchronous notifications [see section **6**].

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

The host can determine the size of notification message packet Input Reports by looking at the HID report descriptor. Notification message packet reports are generally 63 bytes long. If a notification message can't fit into one packet, the device sends multiple packets, each containing the notification message packet format in **Table 6-2** and partial notification message data.

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

1) Knowing from section **2.1.1** that the device uses usage page `0xFF00`, and knowing from the "Where to Find Value" column in the first table of section **6.1** that the desired data is found in the usage with Usage ID `0x0020`, call the platform's USB SDK to retrieve the data from usage `0xFF000020` in the input report.
2) Interpret the blob of data from that Usage according to the second table in section **6.1**.

2.2 How to Use UART and RS-232 Connections (RS-232 Only | UART Only)

When the device is communicating with the host via a serial connection [RS-232 UART or logic level UART], it uses one of two formats. If the device is physically connected to a USB port on the host system, it will communicate with the host through the USB. If it is physically connected to an RS-232 port on the host, it talks to the host over the RS-232 interface.

See section **1.4 About Connections and Data Formats** to determine which of these format rules the device uses:

- Devices identified in **Table 1-1 - Device Connection Types / Data Formats** as using SLIP format use that format to send and receive commands and send command responses. See section **3.2 How to Use SLIP Format**.

3 Data Formats

3.1 How to Use HID Format (HID Only)

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

3.2 How to Use SLIP Format (SLIP Only)

When the host and device exchange data using SLIP format, all messages are composed of a series of binary values between 0x00 and 0xFF.

The SLIP format is defined in Part D, Section 3 of ***Specification of the Bluetooth System, Host Controller Interface, Volume 4***, which is available at https://www.bluetooth.org/Technical/Specifications/adopted.htm. Note the reference to bluetooth.org is intentional, and the specification does indeed apply to other device connection types.

Host software should begin and end commands with SLIP's frame delimiter **C0**, and must take into account SLIP escape sequences that deal with occurrences of C0 inside the SLIP data frame:

- If outbound data contains the byte value **C0**, software should encode it into SLIP as **DB DC**; if inbound SLIP data contains the byte sequence **DB DC**, software should decode it to **C0**.
- If outbound data contains the byte value **DB**, software should encode it into SLIP as **DB DD**; if inbound SLIP data contains the byte sequence **DB DD**, software should decode it to **DB**.

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

3.2.1 Device-Initiated Messages In SLIP Format

When using SLIP format, the device may send **Notification Messages Sent from Device to Host (Extended Notifications Only)** in either normal or Run-Length Encoded (RLE) compressed format, depending on whether RLE would help compress the data or not. The host software should understand both formats. The first byte of the incoming message is a message type field, which indicates what type of data the device is sending and whether the data is RLE compressed, as follows:

- 0x02 = **Notification Uncompressed**, which indicates the Notification Data contains an uncompressed notification message [see section **6 Notification Messages Sent from Device to Host (Extended Notifications Only)**].
- 0x03 = **Notification RLE**, which indicates the Notification Data contains a run-length-encoded compressed notification message [see section **6 Notification Messages Sent from Device to Host (Extended Notifications Only)** and the information below about RLE decoding].

The device implements RLE encoding of Notification Message data as follows:

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

For example, the data `0x44 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x055 0x66 0x00 0x00` is encoded as `0x44 0x55 0x55 0x09 0x66 0x00 0x00 0x02`. A run of 260 `0x00` bytes would be encoded as `0x00 0x00 0xFF 0x00 0x00 0x05`.

3.2.1.1 Notification Messages In SLIP Format (Extended Notifications Only)

Uncompressed **Notification Messages Sent from Device to Host (Extended Notifications Only)** are wrapped in the following block:

Table 3-1 - SLIP Format Notification Uncompressed Wrapper

Bit	7	6	5	4	3	2	1	0
Byte 0	SLIP frame delimiter = 0xC0							
Byte 1	Message Type = 0x02 Notification Uncompressed							
Bytes 2..3	Length of Notification Message field, in big endian order							
Bytes 4..n	Notification Message as defined in section **6.1 About Notification Messages**							
Byte n+1	SLIP frame delimiter = 0xC0							

RLE encoded **Notification Messages Sent from Device to Host (Extended Notifications Only)** are wrapped in the following block:

Table 3-2 - SLIP Format Notification RLE Wrapper

Bit	7	6	5	4	3	2	1	0
Byte 0	SLIP frame delimiter = 0xC0							
Byte 1	Message Type = 0x03 = Notification RLE							
Bytes 2..3	Length of uncompressed Notification Message field, in big endian order							
Bytes 4..n	RLE encoded Notification Message as defined in section **6.1 About Notification Messages**							
Byte n+1	SLIP frame delimiter = 0xC0							

3.2.2 Commands and Responses In SLIP Format

When the device and host are using SLIP format for commands and responses, the host software should wrap all commands in the following block:

Table 3-3 - SLIP Format Command Request Wrapper

Bit	7	6	5	4	3	2	1	0
Byte 0	SLIP frame delimiter = 0xC0							
Byte 1	Message Type = 0x05 Command Request							
Bytes 2..3	Length of Command Request Message field, in big endian order							
Bytes 4..n	Command Request Message from Table 7-1 in section **7.1 About Commands**							
Byte n+1	SLIP frame delimiter = 0xC0							

The device wraps all command responses in the following block:

Table 3-4 - SLIP Format Command Response Wrapper

Bit	7	6	5	4	3	2	1	0
Byte 0	SLIP frame delimiter = 0xC0							
Byte 1	Message Type = 0x04 Command Response Normal, which indicates the payload contains an uncompressed command response message.							
Bytes 2..3	Length of Command Response Message field, in big endian order							
Bytes 4..n	Command Response Message from **Table 7-2** in section **7.1 About Commands**							
Byte n+1	SLIP frame delimiter = 0xC0							

4 Security Levels

Devices can be configured to operate at different Security Levels, which affects the host software's ability to modify **Properties**, and the host software's ability to execute certain **Commands**. The Security Level can be increased by sending commands to the device, but can never be decreased. The sections below provide details about how each security level affects device behavior.

4.1 About Message Authentication Codes (MAC)

Commands in this manual that are tagged "MAC" are **privileged commands**. If the device is set to a Security Level higher than **Security Level 2**, the host software must calculate and append a four-byte Message Authentication Code ("MAC") to the Data field of the message, extending the length of the field by 4 bytes, to prove the sender is authorized to execute that command. If the device is set to **Security Level 2**, the device ignores the MAC field and the Device Serial Number field and the host can set them to all zeroes. If a MAC is required but not present or incorrect, the device returns `0x07`. (Fixed Key Only) If the device is configured to use fixed key encryption instead of DUKPT using **Property 0x6B - Key Management Scheme (Fixed Key Only)**, then MACing is not ever required.

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

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

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

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

4.2 Security Level 2

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

4.3 Security Level 3

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

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

4.4 Command Behaviors By Security Level

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

- **Y** means the command can run at the specified security level.
- **N** means the command is prohibited at the specified security level.
- **C** means the customer may specify **Y** or **S** for that command when ordering.
- **S** means the command is secured [may require MACing, see section **4.1 About Message Authentication Codes (MAC)**]. (Fixed Key Only) If the device is configured to use fixed keys instead of DUKPT key management using **Property 0x6B - Key Management Scheme (Fixed Key Only)**, commands marked with **S** do not require MACing.
- * indicates **Command 0x02 - Reset Device** has special behavior. If an Authentication sequence has failed, only a correctly MACed **Command 0x02 - Reset Device (MAC)** can be used to reset the device. This is to prevent a dictionary attack on the keys and to minimize a denial of service (DoS) attack.

Table 4-1 - Command Behaviors At Each Security Level

Command	Level 2	Level 3
Any command that is not listed in this table works the same at all Security Levels.	Y	Y
Command 0x01 - Set Property (MAC)	Y	S
Command 0x02 - Reset Device (MAC)	Y	*
Command 0x15 - Get / Set Security Level (MAC)	S	S
Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)	N	Y
Extended Command 0x0302 - Cardholder Selection Result	N	Y
Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)	N	Y
Extended Command 0x0304 - Cancel Transaction (EMV Only)	N	Y
Extended Command 0x0305 - Modify Terminal Configuration (MAC)	N	S
Extended Command 0x0307 - Modify Application Configuration (MAC)	N	S
Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)	N	S
Extended Command 0x030C - Set Date and Time (MAC)	N	S
Extended Command 0x030E - Commit Configuration	N	Y
Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)	N	S

5 Encryption, Decryption, and Key Management

5.1 About Encryption and Decryption

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

5.2 How to Determine the Key

When the device and the host are using TDES DUKPT key management [see **Property 0x6B - Key Management Scheme (Fixed Key Only)**] and the device is encrypting data (see **Security Levels**), the host software must do the following to generate a key (the "derived key") to use for decryption:

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

(Fixed Key Only)
As an alternative to TDES DUKPT key management, the device can also be configured to allow the host to manage keys by changing **Property 0x6B - Key Management Scheme (Fixed Key Only)** to use fixed keys. In this case, the host must load fixed keys using **Command 0x4E - Load Fixed Key (Fixed Key Only)** and keep track of which key is currently loaded. All operations that would ordinarily use DUKPT then used fixed keys instead.

The device can be set to require proof that the host knows the current key before it allows the host to load a new fixed key.

The device ships with the following defaults:

- **Initial Fixed Key**: `0000000000000000` (16 zeroes)
- **Initial Fixed Key Key Serial Number (KSN)**: `0000000000` (10 zeroes)
- **Initial Fixed Key Key Check Value (KCV)**: `0x8CA64D`

5.3 How to Decrypt Data

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

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

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

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

6.1 About Notification Messages

This section provides detail about unsolicited generic notification messages the device sends to the host, Each subsection is tagged with the features, connection types, and data formats for which it is relevant.

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

How the host interprets incoming packets to find the data detailed in this section depends on the connection type (see section **2 Connection Types**) and the data format (see section **3 Data Formats**). All packets arrive at the host in the format-dependent structure shown in **Table 6-1**. Each incoming packet can be interpreted using **Table 6-2**. The **notification message** can be interpreted by first assembling all packets pertaining to the notification message, then looking up the corresponding **Notification Identifier** in the sections that follow.

Table 6-1 - How Notification Message Packets Arrive

Format	Where to Find Value
HID	Report identifier 2, Usage identifier 0x20

Table 6-2 - Structure of Packets That Form a Notification Message

Offset	Field Name	Description
0..1	Partial Data Length	The length of the **Data** field contained in the current message. This field is in big endian format. If this value is not equal to the **Complete Data Length**, the device is sending the notification using multiple packets.
2..3	Data Offset	The offset position in bytes within the entire assembled notification where the first byte of the current packet's **Data** field is located. This field is in big endian format. The first byte of the entire notification's Data is at offset zero.
4..5	Notification Identifier	The type of notification being sent. This field is in big endian format. The value corresponds to the notification identifier numbers in the headings of the subsections of section **6 Notification Messages Sent from Device to Host (Extended Notifications Only)**. In many cases, two-byte notification identifiers are assigned such that the high byte indicates a group of related commands, and the low byte specifies a command within that group.
6..7	Complete Data Length	The total length of data for the entire notification message, summing all **Partial Data Lengths** for multiple packets. This field is in big endian format. If this value is not equal to the **Partial Data Length** of the current packet, the device is sending the data using multiple packets.
8..n	Data	May contain part or all of the notification data. The size of this field is contained in the **Partial Data Length** field.

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

Notification Group 0x03 is reserved for EMV L2 notifications that support **Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)**. For more information about the general flow of EMV transactions, see section **7.4 Command Group 0x03 - EMV L2 (EMV Only, Extended Commands Only)**.

6.2.1 Notification 0x0300 - Transaction Status / Progress Information

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

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

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

Notification Data

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

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

6.2.2 Notification 0x0301 - Display Message Request

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

Notification Data

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

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

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

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

6.2.4 Notification 0x0303 - ARQC Message

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

Table 6-3 - Notification Data, ARQC Message

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

6.2.5 Notification 0x0304 - Transaction Result Message

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

Table 6-4 - Notification Data, Transaction Result Message

Offset	Field Name	Value
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.
1	Data Length	Two byte binary, most significant byte first. This gives the total length of the Data message that follows, excluding padding and CBC-MAC.
3	Data	See Appendix **B.3 Transaction Result Messages** and **Property 0x68 – EMV Message Format**. It is expected that the host will save this data as a record of the transaction.

6.3 Notification Group 0x04 - Auxiliary UART (Auxiliary Ports Only) (mDynamo Gen I Only)

Notification Group 0x04 is reserved for Auxiliary UART port notifications that support **Command Group 0x04 - Auxiliary UART (Auxiliary Ports Only, Extended Commands Only)**.

6.3.1 Notification 0x0400 - Auxiliary UART Received Data

The device sends this notification to pass data to the host that it has received from an external UART device via the auxiliary UART port. For information about the auxiliary UART port, see section **7.5 Command Group 0x04 - Auxiliary UART (Auxiliary Ports Only, Extended Commands Only)**.

Notification Data

Offset	Field Name	Value
0	Port Identifier	The identifier of the port. Always set to zero.
1..n	Received Data	The data received. A message may be received over multiple notification messages depending on its length and time between each byte sent.

6.4 Notification Group 0x05 - Auxiliary SPI (Auxiliary Ports Only)

Notification Group 0x05 is reserved for Auxiliary SPI port notifications that support **Command Group 0x05 - Auxiliary SPI (Auxiliary Ports Only, Extended Commands** Only).

6.4.1 Notification 0x0500 - Auxiliary SPI Data Change

The device sends this notification to inform the host that an external SPI device has changed the data on the auxiliary SPI port. For information about the auxiliary SPI port, see section **7.6 Command Group 0x05 - Auxiliary SPI (Auxiliary Ports Only, Extended Commands** Only).

Notification Data

Offset	Field Name	Value
0	Port Identifier	The identifier of the port. Always set to zero.
1	Data	The data byte. The data byte indicates what data changed. See the Data Field table below for information about interpreting the value of this field.

Data Field (Bit 0 is the least significant bit)

Bit	Field Name	Value
0	DAV Low	This bit is set high when the DAV (Data available) input signal from the connected SPI device transitions from high to low. **Property 0x6A - Auxiliary SPI Configuration (Auxiliary Ports Only)**, bit DAV Notify Low, can be used by the host to enable/disable this notification.
1	DAV High	This bit is set high when the DAV (Data available) input signal from the connected SPI device transitions from low to high. **Property 0x6A - Auxiliary SPI Configuration (Auxiliary Ports Only)**, bit DAV Notify High, can be used by the host to enable/disable this notification.
2..7	Reserved	These bits should be ignored.

7 Commands

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

7.1 About Commands

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

Table 7-1 - Command Request Message (Host Sends to Device to Initiate a Command)

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

Command Number is a one byte value that contains the requested command number. Section **7 Commands** lists all available commands.

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

Command Request Data contains command data as defined in the documentation for the selected command in section **7 Commands**.

Table 7-2 - Command Response Message (Host Sends to Device to Retrieve Data or Responses)

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

Result Code is a one-byte value the device sends to indicate success or the failure mode of the command. Section **7.2 About Result Codes** provides more detail.

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

Command Response Data contains response data as defined in the documentation for the selected command in section **7 Commands**.

7.2 About Result Codes

There are two types of **Result Code** values the device can return in its response: **Generic** result codes (listed in **Table 7-3**), which have the same meaning for all commands, and **command-specific** result codes, which can have different meanings for different commands, and are listed with every command in this section. Generic result codes always have the most significant bit set to zero, and command-specific result codes always have the most significant bit set to one.

Table 7-3 - Generic Result Codes

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

7.3 General Commands

7.3.1 Command 0x00 - Get Property

This command gets a property from the device. For details about properties, see section **8 Properties**.

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

Table 7-4 - Request Data for Command 0x00 - Get Property

Data Offset	Value
0	Property ID

Table 7-5 - Response Data for Command 0x00 - Get Property

Data Offset	Value
0..n	Property Value

Property ID is a one-byte value that identifies the property. A full list of properties can be found in section **8 Properties**.

Property Value consists of the multiple-byte value of the property. The number of bytes in this value depends on the type of property and the length of the property. **Table 7-6** describes the available property types.

Table 7-6 - Property Types

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

The result codes for the **Get Property** command can be any of the generic result codes listed in **Table 7-3** on page **42**.

7.3.2 Command 0x01 - Set Property (MAC)

This command sets a property in the device. For security purposes, this command is privileged. When the Security Level is set to higher than 2 (see section **4 Security Levels**), this command must be MACed to be accepted [see section **4.1 About Message Authentication Codes (MAC)**]. The command is logically paired with **Command 0x00 - Get Property**. For details about properties, see section **8 Properties**.

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

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

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

Response Data: None

The result codes for the **Set Property** command can be any of the generic result codes listed in **Table 7-3** on page **42**. If the **Set Property** command gets a result code of `0x07`, it means the required MAC was absent or incorrect.

Property ID is a one-byte value that identifies the property. A full list of properties can be found in section **8 Properties**.

Property Value consists of multiple bytes containing the value of the property. The number of bytes in this value depends on the property. **Table 7-4** describes the available property types.

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

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

7.3.3 Command 0x02 - Reset Device (MAC)

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

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

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

Request Data Field: None

Response Data Field: None

Result codes:
0x00 = Success
0x07 = Incorrect MAC, or authentication sequence is pending

Example Request (Hex)

Cmd Num	Data Len	Data
02	00	

Example Response (Hex)

Result Code	Data Len	Data
00	00	

7.3.4 Command 0x09 - Get Current TDES DUKPT KSN

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

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

Request Data: None

Table 7-9 - Response Data for Command 0x09 - Get Current TDES DUKPT KSN

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

Result codes:
0x00 = Success
0x02 = Bad Parameter - The Data field in the request is not the correct length. The request command contains no data, so the Data Length must be 0.

Example Request (Hex)

Cmd Num	Data Len	Data
09	00	None

Example Response (Hex)

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

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

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

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

(Fixed Key Only)
If the device is configured to use fixed key encryption using **Property 0x6B - Key Management Scheme (Fixed Key Only)**, then MACing is not required. In this case, the MAC field can be omitted.

Table 7-10 - Request Data for Command 0x15 - Get / Set Security Level (MAC)

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

Table 7-11 - Response Data for Command 0x15 - Get / Set Security Level (MAC)

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

Result codes:
0x00 = Success
0x02 = Bad Parameters. The Data field in the request is not a correct length OR the specified Security Level is invalid; OR the current Security Level is 4.
0x07 = Incorrect MAC; command not authorized

Example Set Security Level Request (Hex)

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

Example Set Security Level Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Data
15	00	

Example Get Security Level Response (Hex)

Result Code	Data Len	Data
00	01	03

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

The host uses this command to send **extended commands** to the device as one or more data packets. This **extended commands protocol** doubles the command number namespace to two bytes, doubles the result code namespace to two bytes, and supports commands and responses which require larger data payloads than those available for standard commands (shown in **Table 7-1** and **Table 7-2** in section **7.1 About Commands**).

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

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

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

Table 7-12 - Request Data for Command 0x49 - Send Extended Command Packet (Extended Commands Only)

Offset	Field Name	Description
0..1	Extended Data Offset	This field is in big endian format. It indicates the byte offset position of this packet's Extended Data field, relative to the complete extended data field being sent as multiple packets. The Extended Data Offset of Packet 0 is 0.
2..3	Extended Command Number	This field is in big endian format and contains the number of the command to execute. For one-byte command numbers, the high byte should be set to zero.
4..5	Complete Extended Data Length	This field is in big endian format and gives the total length of the Extended Data field the host is sending as multiple packets.

6..n	Extended Data	This field contains either part or all of the extended data request the host is sending to the device. The size of this Extended Data field can be determined by subtracting the Extended Data field's offset within the request (6) from the request's total data length (N). In most cases the request's complete data payload can have a maximum value of 58 (for example see section **2.1.2 How to Send Commands On the USB Connection**), so this field can have a maximum length of 58 - 6 = 52 bytes.

Table 7-13 - Response Data for Command 0x49 - Send Extended Command Packet (Extended Commands Only)

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

Result Codes:
See command description.

Example Request (Hex)

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

Example Response (Hex)

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

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

The host uses this command to retrieve additional response data longer than the current connection type's maximum packet size. After calling a command, if the device returns generic result code **0x0A Extended response first packet** (see **Table 7-3 - Generic Result Codes** on page **42**), the host software should begin buffering the complete Extended Response starting with the initial response, then call this command repeatedly until it has retrieved the complete Extended Response.

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

Request Data: None

Table 7-14 - Response Data for Command 0x4A - Get Extended Response (Extended Commands Only)

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

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

Example Request (Hex)

Cmd Num	Data Len	Data
4A	00	

Example Response (Hex)

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

7.3.8 Command 0x4E - Load Fixed Key (Fixed Key Only)

Caution: Make sure the host knows the current fixed key before calling this command with the Authentication Required field set to True, or there will be no way to load a new fixed key onto the device.

The host software uses this command to load a 16-byte fixed key into the device when **Property 0x6B - Key Management Scheme (Fixed Key Only)** is set to **Fixed Key**.

To load a fixed key, the host software should follow these steps:

1) Generate or obtain the key to be injected. This can be as simple as generating a string of 16 random hexadecimal digits, or as involved as requesting a key management team generate a key in a specific format using a Hardware Security Module (HSM). Specific methods of generating keys are outside the scope of this documentation.
2) Decide whether to load a single encrypted fixed key or send it as two components in the clear. If the key is a simple string of 16 hexadecimal digits but the host is loading it as two components in the clear, one component can be the actual key and the other can be `0000000000000000` (16 zeroes).
3) Decide on a 10-byte KSN for the key. The device includes this KSN alongside any encrypted data it sends [for example in data object `DFDF56` in **ARQC Messages (EMV Only)** and **Transaction Result Messages (EMV Only)**], so the host can identify the key it should use to decrypt. Choice of KSN is completely the purview of the host software. For example, implementers may choose to use a fixed KSN (effectively no KSN) for closed systems, or may start at `0000000000` (10 zeroes) and increment on each call to this command, or any other algorithm that fits the chosen implementation.
4) Know the current fixed key value, if the device is set to require authentication. When the device ships, the intial fixed key is set to `0000000000000000` (16 zeroes).
5) Know if the device is configured to require authentication before loading a fixed key. When the device ships, it is configured to not require authentication so the host can load the first non-default fixed key, and when the host loads that key, it can use the **Authentication Required** flag in this command to require authentication for future fixed key load operations.
6) If loading a single encrypted key:
   a) Encrypt the key using the current fixed key and Electronic Codebook (ECB) encryption
   b) If the device is configured to require authentication, successfully complete the authentication sequence of **Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)** followed by **Command 0x50 - Fixed Key Authentication Response (Fixed Key Only)**.
   c) Send this command with Key Data Type set to Encrypted Fixed Key, the chosen KSN, the encrypted Key Data, and the desired Authentication Required setting for subsequent fixed key load operations.
7) If loading two components in the clear:
   a) If the device is configured to require authentication, successfully complete the authentication sequence of **Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)** followed by **Command 0x50 - Fixed Key Authentication Response (Fixed Key Only)**.
   b) Send this command with Key Data Type set to Clear Component 1, KSN null, the first key component in Key Data, and null in Authentication Required.
   c) Within 2 minutes, send this command again with Key Data Type set to Clear Component 2, the chosen KSN, the second key component in Key Data, and the desired Authentication Required setting for subsequent fixed key load operations.

The host can also send this command with a request data length of zero; the device does not load any key, and returns information about the current fixed key in its response.

Changes to fixed key settings are effective immediately and persist after a power cycle or reset.

Table 7-15 - Request Data for Command 0x4E - Load Fixed Key (Fixed Key Only)

Offset	Field Name	Description
0	Key Data Type	0 = Encrypted Fixed Key 1 = Clear Component 1. The Key Data field contains component 1 of the new fixed key in the clear. 2 = Clear Component 2. The Key Data field contains component 2 of the new fixed key in the clear. If sent successfully, the device XORs the components to form the new fixed key.
1..10	Key Serial Number	10 byte Key Serial Number (KSN) The device ignores this field when the key data type field is set to 1 (Clear Component 1), because the host also sends this field for Clear Component 2.
11..26	Key Data	16 byte key data, formatted as defined in the Key Data Type field.
27	Authentication Required	0x00 = Authentication not required the next fixed key load 0x01 = Authentication required for the next fixed key load The device ignores this field when the key data type field is set to 1 (Clear Component 1), because the host also sends this field for Clear Component 2.

Table 7-16 - Response Data for Command 0x4E - Load Fixed Key (Fixed Key Only)

Offset	Field Name	Description
0..9	Key Serial Number	See this field's description in the request data.
10..12	Key Check Value (KCV)	The KCV can be used to help identify/validate what key is loaded. It contains the first 3 bytes of the result of encrypting an 8 byte field of zeroes with the current fixed key.
13	Authentication Required	See this field's description in the request data.

Result codes:
0x00 = Success
0x01 = Failure
0x02 = Bad Parameter

Example Request (Hex)

Cmd Num	Data Len	Data
4E	1C	00 0102030405060708090A 8CA64DE9C1B123A78CA64DE9C1B123A7 00 (New key of all zeroes is encrypted under the current key of all zeroes)

Example Response (Hex)

Result Code	Data Len	Data
00	0E	01 02 03 04 05 06 07 08 09 0A 8C A6 4D 00

7.3.9 Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)

The host uses this command, along with **Command 0x50 - Fixed Key Authentication Response (Fixed Key Only)**, to authenticate before calling **Command 0x4E - Load Fixed Key (Fixed Key Only)** if that command has previously required authentication to load subsequent fixed keys. Performing this sequence successfully proves to the device that the host has knowledge of the current fixed key value.

Request Data: None

Table 7-17 - Response Data for Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)

Offset	Field Name	Description
0 - 7	Authentication Challenge	An 8-byte random number encrypted using the current fixed key.

Result codes:
0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
4F	00	

Example Response (Hex)

Result Code	Data Len	Data
00	08	5B0BF27FCDB6C280 (Random number 371B5A89B509B5FD is encrypted under the current key of all zeroes

7.3.10 Command 0x50 - Fixed Key Authentication Response (Fixed Key Only)

The host uses this command, along with **Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)**, to authenticate before calling **Command 0x4E - Load Fixed Key (Fixed Key Only)** if that command has previously configured the device to require authentication for loading subsequent fixed keys. Performing this sequence successfully proves to the device that the host has knowledge of the current fixed key value.

The host should first call **Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)**. The device encrypts a random string using the current fixed key, and sends the resulting encrypted 8-byte challenge in its response to the host. The host should then decrypt the challenge using the current fixed key, and send the first four bytes back to the device using this command.

If the Authentication Response field matches the first 4 bytes of the random string the device used, authentication succeeds, otherwise it fails. The result code indicates if it succeeded or failed.

If successful, the device remains in authenticated mode until it is power cycled/reset or until it receives another **Command 0x4F - Fixed Key Authentication Challenge (Fixed Key Only)**.

Table 7-18 - Request Data for Command 0x50 - Fixed Key Authentication Response (Fixed Key Only)

Offset	Field Name	Description
0 - 3	Authentication Response	First four bytes of the unencrypted Authentication Challenge

Response Data: None

Result codes:
0x00 = Success
0x01 = Failure
0x02 = Bad Parameter

Example Request (Hex)

Cmd Num	Data Len	Data
50	04	371B5A89 (Challenge 5B0BF27FCDB6C280 is decrypted under the current key of all zeroes)

Example Response (Hex)

Result Code	Data Len	Data
00	00	

7.3.11 Command 0x51 - External LED Control (External LED Control Only)

The host uses this command to directly control the External LED Connector. After a power cycle or reset, the device defaults to driving the external LED and the on-board General Status LED identically. This command only affects the external LED and does not affect the on-board LED.

If the host includes data in the command request message, the device drives the external LED as the command specifies, and does not include data in the response. If the host does not include data in the request message, the device returns the current LED control settings in the response.

See **Table 1-2** External LED feature for information about which devices support this feature.

Table 7-19 - Request Data for Command 0x51 - External LED Control (External LED Control Only)

Offset	Field Name	Description
0	State	One byte specifying what the external LED should do: 0 = Firmware application controlled (default) 1 = Off 2 = Green 3 = Red 4 = Amber
1	Blink Period	One byte specifying the blink period in 10ms units. If the blink period is set to 0x00, the LED does not blink, otherwise the LED continuously turns on for the blink period then off for the blink period. For example, if the blink period is set to 5, the LED turns on for 50ms then off for 50ms then repeats. If the **State** field is set to **Firmware application controlled** or **Off**, the host should set Blink Period to 0x00 and the device ignores it.

Table 7-20 - Response Data for Command 0x51 - External LED Control (External LED Control Only)

Offset	Field Name	Description
0	State	See request data field description.
1	Blink Period	See request data field description.

Result codes: 0x00 = Success

Example Request (Hex)

Cmd Num	Data Len	Data
51	02	03 00

Example Response (Hex)

Result Code	Data Len	Data
00	00	

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

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

7.4.1 About MACs

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

7.4.2 About EMV L2 Transaction Flows (EMV Only)

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

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

i) The device sends the host **Notification 0x0302 - Cardholder Selection Request** to prompt the cardholder to **Select Application** with a list of available applications, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report **Waiting for Cardholder Response / Waiting for Cardholder Application Selection**.

ii) After the cardholder selects an application, the host passes the selection to the device by sending **Extended Command 0x0302 - Cardholder Selection Result**.

c) The device sends **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Initiating Application**.

8) (Contact Only) If the cardholder has inserted a chip card, and the card's selected application reports to the device that the cardholder should select a language, the device sends the host **Notification 0x0302 - Cardholder Selection Request** to prompt the cardholder to **Select Language** with a list of available languages, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report event **Waiting for Cardholder Response / Waiting for Cardholder Language Selection**. After the cardholder selects a language, the host passes the selection to the device by sending **Extended Command 0x0302 - Cardholder Selection Result**.

9) The device initiates communication with the card and sends the host **Notification 0x0300 - Transaction Status / Progress Information** reporting **Transaction Progress Change/ Reading Application Data**. If an error or other type of failure occurs during this step, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0301 - Display Message Request** to the host with the message **TRANSACTION TERMINATED**, followed by **Notification 0x0304 - Transaction Result Message**.

10) Depending on the capabilities of the card and the device, the device authenticates the card data using SDA, DDA, or CDA. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Offline Data Authentication**.

11) The steps from here through **Card Action Analysis** below are collectively referred to as the **Risk Management** process.

12) The device checks to make sure the selected application is valid for the transaction, and is compatible with the device (such as application version number, application usage control, and application effective / expiration date), and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Process Restrictions**.

13) The device uses the cardholder verification related data in the card or contactless payment device to determine which cardholder verification method (CVMs) to use. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Cardholder Verification**.

14) The device performs terminal risk management procedures, which involves floor limit checking, velocity checking, and periodically forcing online authorization to protect against fraud, and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Terminal Risk Management**.

15) The device analyzes the results of the previous steps and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Terminal Action Analysis**.

16) The device rolls up the results of the previous Risk Management process:

a) If the Risk Management process encounters an error or determines the transaction or payment method fails to meet required criteria, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0301 - Display Message Request** to the host with the message

TRANSACTION TERMINATED, followed by **Notification 0x0304 - Transaction Result Message**, and terminates the transaction

b) If the Risk Management process determines the transaction is too risky to approve, the device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Data Error / Transaction Error**, followed by **Notification 0x0304 - Transaction Result Message**, followed by **Notification 0x0301 - Display Message Request** with message **DECLINED** to notify the cardholder, and terminates the transaction.

17) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Generating First Application Cryptogram**. The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** reporting **Transaction Progress Change/ Card Action Analysis**.

18) If the device is NOT configured as Online-Only Terminal Type [see **Appendix C EMV Terminal and Application Settings (EMV Only)**] and the Risk Management processes determined the transaction is OK to perform offline, the device reports the transaction result to the host as follows:

a) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**.

b) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction** and either **Transaction Approved** or **Transaction Declined**, then sends the host **Notification 0x0301 - Display Message Request** with message **APPROVED** or **DECLINED** to notify the cardholder.

c) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.

19) If the device is configured as an Online Only Terminal Type [see **Appendix C EMV Terminal and Application Settings (EMV Only)**] or the Risk Management processes determined the transaction must be performed online, the device reports the transaction result to the host as follows:

a) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Online Processing**, followed by **Notification 0x0303 - ARQC Message**.

b) The next event depends on whether the device supports the Contact Quick Chip feature or Contactless Quick Chip feature (see **Table 1-2**) and whether the host specified Quick Chip as an Option when it started the transaction:

c) If Quick Chip operation is supported and in effect:

i) The device immediately constructs its own internal ARPC Response, with tag 8A set to 'Z3' to coordinate the transaction with the card or other payment method, and sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**, followed by **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction / Transaction Declined**.

ii) The device sends the host **Notification 0x0301 - Display Message Request** with message **REMOVE CARD** to notify the cardholder the card can be removed.

iii) The host should then process the ARQC Message data, including replacing the default amount with the final transaction amount, and should coordinate with the transaction processor to retrieve a final transaction result. Because in this case the device is not involved in determining the final transaction result, it does not send a notification to the host to show **APPROVED** or **DECLINED**. Instead, the host should display an appropriate message (such as **QUICK CHIP APPROVED** / **QUICK CHIP DECLINED**) to the cardholder based on the final transaction result.

iv) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.

d) If Quick Chip operation is NOT supported or is not in effect:

i) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change/ Waiting for Online Processing Response**.

ii) The host processes the ARQC Message data and uses it to coordinate with the transaction processor to receive an ARPC Response, which it processes and sends to the device using **Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)**. (Contact Only) Alternatively, the host may implement host-driven Quick Chip by instead constructing its own preliminary ARPC Response with tag 8A set to 'Z3' and sending it to the device immediately, without waiting for a transaction processor response. The device responds by sending **Notification 0x0301 - Display Message Request** to the host with message **DECLINED** and ending the transaction. The host should suppress this message and take over the remainder of the transaction, including notifying the cardholder to remove the card, determining the final transaction amount, coordinating with the transaction processor to retrieve a final transaction result, and interacting with the cardholder.

iii) The device communicates with the chip card to determine whether to approve or decline the transaction, then sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **Transaction Progress Change / Transaction Complete**.

iv) The device sends the host **Notification 0x0300 - Transaction Status / Progress Information** to report **End of Transaction** and either **Transaction Approved** or **Transaction Declined**, then and sends the host **Notification 0x0301 - Display Message Request** with message **APPROVED** or **DECLINED** to notify the cardholder of the transaction result.

v) The device ends the transaction by sending the host **Notification 0x0304 - Transaction Result Message**, which contains transaction details the host should save for later verification. The transaction result message indicates whether the host must prompt the cardholder to provide a signature.

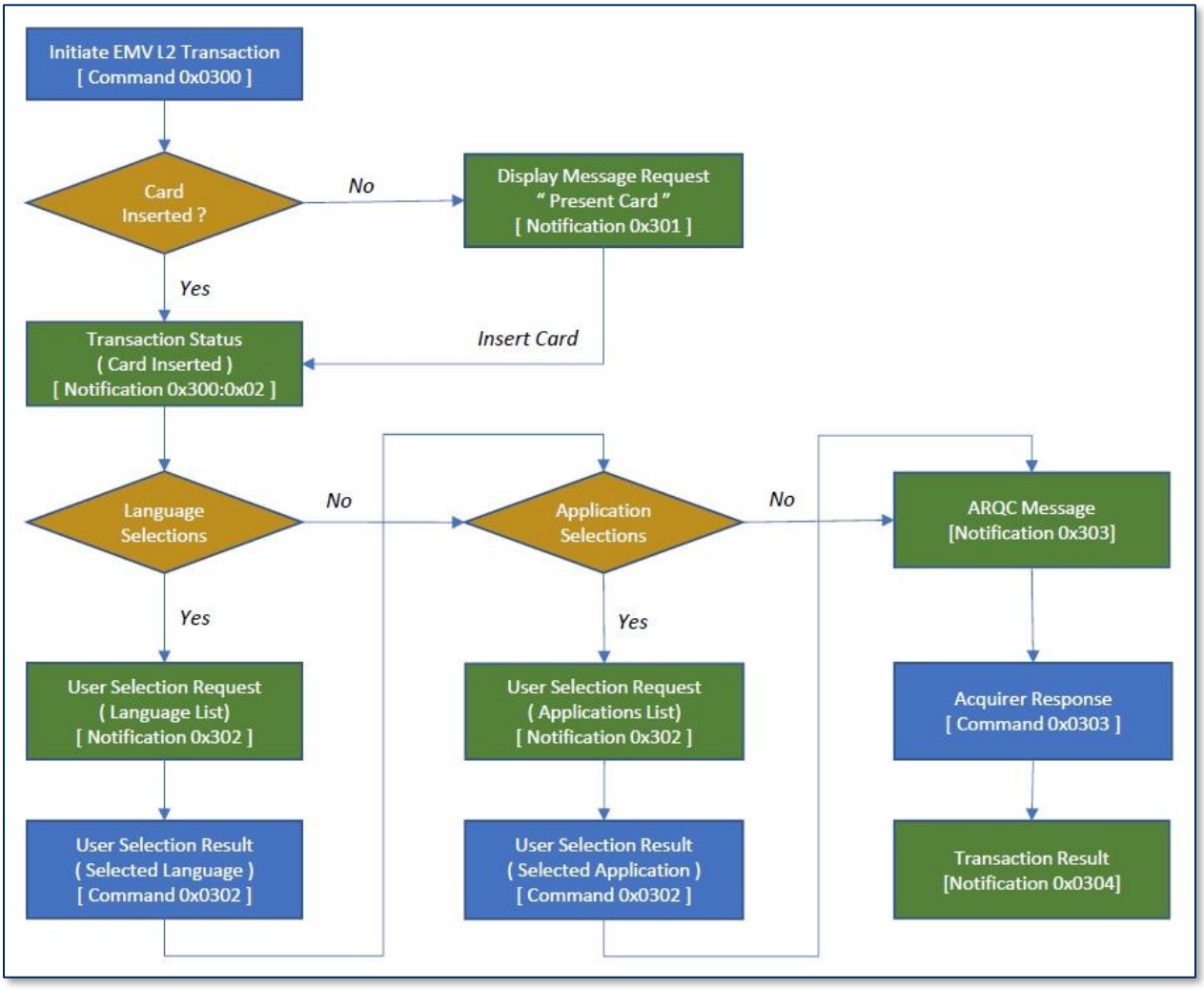

Figure 7-1 - Simplified EMV Transaction Flow

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

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

This command is used to initiate an EMV transaction sequence, which (in a typical no-error case) flows as described in section **7.4.2 About EMV L2 Transaction Flows (EMV Only)**. The command provides all the data the device needs to start the transaction, and the device returns a response to the host to indicate whether the transaction will proceed. If the input fields for the command are not formatted correctly and within defined limits, the response message returns an error code indicating why the command could not proceed. If the device is set to a lower security level than **Security Level 3**, the device refuses this command, unless the device is an mDynamo, which accepts this command at **Security Level 2**.

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

The device's system date and time must be set prior to sending this command. Devices without a battery-backed real time clock require the host to set the date and time using **Extended Command 0x030C - Set Date and Time (MAC)** every time the device is power cycled or reset.

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

- **Extended Command 0x0302 - Cardholder Selection Result**
- **Extended Command 0x0303 - Online Processing Result / Acquirer Response**
- **Extended Command 0x0304 - Cancel Transaction (EMV Only)**
- **Extended Command 0x0305 - Modify Terminal Configuration (MAC)**

Table 7-21 - Request Data for Extended Command 0x0300 - Initiate EMV Transaction (EMV Only)

Offset	Field Name	Value
0	Transaction Flow Time	Specifies the maximum time, in seconds, for cardholder interaction events to complete while processing a transaction. Values from 0x01 to 0xFF are allowed (1 to 255 seconds). The timer starts at the beginning of each event. If the cardholder action does not occur within the specified time, the transaction proceeds as follows: • **Cardholder present payment** timeout: The transaction terminates. • (Contact Only) **Cardholder language selection** timeout: The transaction continues with the default language. • (Contact Only) **Cardholder application selection** timeout: The transaction terminates. (OEM Features) A value of 0x00 directs the device to initiate an EMV transaction and wait indefinitely until a cardholder presents payment, or the host issues **Extended Command 0x0304 - Cancel Transaction (EMV Only)** or the device is power cycled or reset. This allows the host to drive a loop in unattended solutions that idle until the next cardholder initiates a transaction by swiping, inserting, or tapping.
1	Card Type to Read	Card Type to Read (OR the following values together): 0x01 = Reserved 0x02 = Contact chip card (Contact Only) 0x04 = Reserved
2	Options	0x00 = Normal 0x01 = Reserved for Bypass PIN 0x02 = Reserved for Force Online (Quick Chip Only | Contactless Quick Chip Only) To use Quick Chip mode, set the most significant bit to '1'. For example: 0x80 = Normal, Use Quick Chip
3..8	Amount Authorized	Amount Authorized (EMV Tag 9F02, format n12, 6 bytes). For Transaction Type **Refund** (0x20), this must contain the refund amount.
9	Transaction Type	0x00 = Purchase (covers transaction types Payment, Goods, and Services) 0x02 or 0x09 = Cash back (0x09 only supported when using contactless) 0x20 = Refund. If the specified Card Type to Read does not formally support refunds, the host can still use **Refund** to retrieve card data it needs to process a refund transaction, but internally and in its responses to the host, the device forces Transaction Type to **Purchase** and replaces Amount Authorized with **0.00**.
10..15	Cash Back	Cash back amount (if non-zero, EMV Tag 9F03, format n12, 6 bytes). For Transaction Type **Refund** (0x20) this must be 0.00.
16..17	Transaction Currency Code	Transaction Currency Code (EMV Tag 5F2A, format n4, 2 bytes) Valid values are the numerical codes from ***ISO 4217 Codes for the representation of currencies***, for example: 0x0000 = Use Selected Application's Currency Code Terminal Setting 0x0840 = US Dollar

		0x0978 = Euro
18	Reporting Option	This single byte field indicates the level of Transaction Status notifications the host wants the device to send during the transaction: 0x00 = Termination status only (normal termination, payment method communication or data error, timeout, host cancel) 0x01 = Major status changes (terminations plus card insertions and waiting for cardholder) (Contact Only) 0x02 = All status changes (documents the entire transaction flow)

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

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

Example Request (Hex)

Header	
Command Number	49
Data Length	19
Data	
Extended Data Offset	0000
Extended Command Number	0300
Complete Extended Data Length	0013
Extended Data	3C020000000000150000000000000000084002

Example Response (Hex)

Header	
Result Code	0A
Data Length	06

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

7.4.4 Extended Command 0x0302 - Cardholder Selection Result

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

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

Table 7-22 - Request Data for Extended Command 0x0302 - Cardholder Selection Result

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

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

Result codes:
0x0000 = Success, the Selection Result was received
0x038B = Invalid Selection Status
0x038C = Invalid Selection Result
0x038D = Failure, no transaction currently in progress

Example Request (Hex)

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

Example Response (Hex)

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

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

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

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

Table 7-23 - Request Data for Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)

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

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

Result codes:
0x0000 = Success, the Selection Result was received
0x038D = Failure, no transaction currently in progress
0x038F = Failure, transaction already in progress

Example Request (Hex)

Header	
Command Number	49
Data Length	39
Data	
Extended Data Offset	0000
Extended Command Number	0303
Complete Extended Data Length	003C
Extended Data	003AF92EDFDF540A00000000000000000000DFDF550182DFDF250F42333545324344303830313136414 1FA0670048A02303000

Example Response (Hex)

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

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

Example Request Following Up For Packet 0 (Hex)

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

Example Response Following Up For Packet 0 (Hex)

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

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

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

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

Request Data: None

Response Data: None

Result codes:
0x0000 = Success, the transaction was cancelled
0x038D = Failure, no transaction currently in progress
0x038F = Failure, transaction in progress, cardholder already presented payment

Example Request (Hex)

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

Example Response (Hex)

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

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

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

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

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

(Fixed Key Only)
If the device is configured to use fixed key encryption using **Property 0x6B - Key Management Scheme (Fixed Key Only)** or the device's security level is less than 3, then MACing is not required. In this case, the Device Serial Number and MAC fields can be set all zeroes.

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

Table 7-24 - Request Data for Extended Command 0x0305 - Modify Terminal Configuration (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1	Slot Number	EMV Terminal Slot Number. Must be 0x01.
2	Operation	0x01 = Write Operation 0xFF = Set to Factory Defaults (sets all items, Terminal, Applications, and Application Public Keys to factory default values)
3	Database Selector	(Contact Only) 0x00 = EMV Contact L2
4..19	Device Serial Number (DSN)	16 Bytes DSN
20..n	Objects To Write	Note: Not needed if Operation is 0xFF Set to Factory Defaults. FA<len> /* container for generic data */ <tag><len><value> … <tag><len><value>
n..n+3	MAC	MAC computed on **Device Serial Number (DSN)** and **Objects to Write** fields. See section **7.4.1 About MACs**.

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

Result codes:
0x0000 = Success
0x0390 = Device Has No Keys

0x0391 = Invalid Device Serial Number
0x0392 = Invalid Type of MAC field
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field
0x0396 = Invalid Objects to Write field
0x0397 = Invalid MAC
0x0399 = Object Write Protected

Example Request (Hex)

Header	
Command Number	49
Data Length	2A
Data	
Extended Data Offset	0000
Extended Command Number	0305
Complete Extended Data Length	0024
Extended Data	00010100nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnFA0A9F160731323334353637xxxxxxxx Where nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn is the 16-byte Device Serial Number (DSN) and xxxxxxxx is the 4-byte MAC

Example Response (Hex)

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

7.4.8 Extended Command 0x0306 - Read Terminal Configuration

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

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

Table 7-25 - Request Data for Extended Command 0x0306 - Read Terminal Configuration

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

Table 7-26 - Response Data for Extended Command 0x0306 - Read Terminal Configuration

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

Result codes:
0x0000 = Success
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field
0x0396 = Invalid Tags to Read field

Example Request (Hex)

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

Example Response (Hex)

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

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

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

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

(Fixed Key Only)
If the device is configured to use fixed key encryption using **Property 0x6B - Key Management Scheme (Fixed Key Only)** or the device's security level is less than 3, then MACing is not required. In this case, the Device Serial Number and MAC fields can be set all zeroes.

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

Table 7-27 - Request Data for Extended Command 0x0307 - Modify Application Configuration (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1	Slot Number	EMV Application Slot Number See **Appendix C EMV Terminal and Application Settings (EMV Only)** to determine how many application slots the device has for the selected database.
2	Operation	0x01 = Write Operation
3	Database Selector	(Contact Only) 0x00 = EMV Contact L2
4..19	Device Serial Number (DSN)	16 Bytes DSN
20..n	Objects to Write	FA<len> /* container for generic data */ <tag><len><value> … <tag><len><value>
n..n+3	MAC	MAC computed on **Device Serial Number (DSN)** and **Objects to Write** fields. See section **7.4.1 About MACs**.

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

Result codes:
0x0000 = Success
0x0390 = Device Has No Keys
0x0391 = Invalid Device Serial Number
0x0392 = Invalid Type of MAC field
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field

0x0396 = Invalid Objects to Write field
0x0397 = Invalid MAC

Example Request (Hex)

Header	
Command Number	49
Data Length	27
Data	
Extended Data Offset	0000
Extended Command Number	0307
Complete Extended Data Length	0021
Extended Data	00010100nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnFA079F1B0400002710xxxxxxxx Where nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn is the 16-byte Device Serial Number (DSN) and xxxxxxxx is the 4-byte MAC

Example Response (Hex)

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

7.4.10 Extended Command 0x0308 - Read Application Configuration

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

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

Table 7-28 - Request Data for Extended Command 0x0308 - Read Application Configuration

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

Table 7-29 - Response Data for Extended Command 0x0308 - Read Application Configuration

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

Result codes:
0x0000 = Success
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field
0x0396 = Invalid Tags to Read field

Example Request (Hex)

Header	
Command Number	49
Data Length	0D

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

Example Response (Hex)

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

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

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

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

(Fixed Key Only)
If the device is configured to use fixed key encryption using **Property 0x6B - Key Management Scheme (Fixed Key Only)** or the device's security level is less than 3, then MACing is not required. In this case, the Device Serial Number and MAC fields can be set all zeroes.

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

Table 7-30 - Request Data for Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)

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

n..n+3	MAC	MAC computed on **Device Serial Number (DSN)** and **Objects to Write** fields. See section **7.4.1 About MACs**.

Table 7-31 - Response Data for Extended Command 0x0309 - Modify Acquirer Public Key CAPK (MAC, EMV ODA Only)

Offset	Field Name	Value
0	Slot Number	When the Next Available slot number (0xFF) is used, this field returns the next available slot used if successful, otherwise this field is omitted.

Result codes:
0x0000 = Success
0x0390 = Device Has No Keys
0x0391 = Invalid Device Serial Number
0x0392 = Invalid Type of MAC field
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field
0x0396 = Invalid Objects to Write field
0x0397 = Invalid MAC
0x0398 = No Slots Available
0x039B = Invalid CAPK Checksum

Example Request (Hex)

Header	
Command Number	49
Data Length	39
Data	
Extended Data Offset	0000
Extended Command Number	0309
Complete Extended Data Length	0033
Extended Data	00010100nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnFA19DFDF790100DFDF7A0100DFDF7B0100DFDF7C0100DFDF7D0100xxxxxxxx Where nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn is the 16-byte Device Serial Number and xxxxxxxx is the 4-byte MAC

Example Response (Hex)

Header	
Result Code	0A
Data Length	06

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

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

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

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

Each CAPK database contains up to 51 key slots, each formatted as shown in **Table 7-32**.

Table 7-32 - Certificate Authority Public Key Slots 1 to 51 Data

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

Table 7-33 - Request Data for Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)

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

Table 7-34 - Request Data for Extended Command 0x030A - Read Acquirer Public Key CAPK (EMV ODA Only)

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

Result codes:
0x0000 = Success
0x0393 = Invalid Slot Number field
0x0394 = Invalid Operation field
0x0395 = Invalid Database Selector field
0x0396 = Invalid Tags to Read field

Example Request (Hex)

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

Example Response (Hex)

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

7.4.13 Extended Command 0x030B - Read EMV Kernel Information

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

This command is used to read kernel information.

Table 7-35 - Request Data for Extended Command 0x030B - Read EMV Kernel Information

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

Table 7-36 - Response Data for Extended Command 0x030B - Read EMV Kernel Information

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

0x0000 = Success
0x0386 = Invalid Mode

Example Request (Hex)

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

Example Response (Hex)

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

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

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

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

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

Devices that do not have a battery-backed real time clock must use this command frequently because (a) the clock must be set before the device can process EMV transactions, and (b) the host software must use this command every time the device is power cycled or reset.

Table 7-37 - Request Data for Extended Command 0x030C - Set Date and Time (MAC)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1.
1..16	Device Serial Number	16 Bytes Device Serial Number. Devices except the following can set this field to all zeroes. • eDynamo with firmware part number 1000003354 revisions earlier than G02 • eDynamo with firmware part number 1000002649
17	Month	Value from 0x01..0x0C
18	Day	Value from 0x01..0x1F (less depending on month)
19	Hour	Value from 0x00..0x17
20	Minute	Value from 0x00..0x3B
21	Second	Value from 0x00..0x3B
22	Unused	Value from 0x00..0x06
23	Year	Value from 0x00 (2008)..0x44 (2076)
24..27	MAC	MAC computed over all preceding fields except **Type of MAC**. Devices except the following can set this field to all zeroes. • eDynamo with firmware part number 1000003354 revisions earlier than G02 • eDynamo with firmware part number 1000002649

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

Result codes:
0x0000 = Success
0x0390 = Device Has No Keys
0x0391 = Invalid Device Serial Number
0x0392 = Invalid Type of MAC field
0x0396 = Invalid Date / Time data
0x0397 = Invalid MAC

Example Request (Hex)

Header	
Command Number	49
Data Length	22
Data	
Extended Data Offset	0000
Extended Command Number	030C
Complete Extended Data Length	001C
Extended Data	00000000000000000000000000000000021C0F380B0009xxxx xxxx Where xxxxxxxx is the 4-byte MAC

Example Response (Hex)

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

7.4.15 Extended Command 0x030D - Read Date and Time

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

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

Request Data: None

Table 7-38 - Response Data for Extended Command 0x030D - Read Date and Time

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

Result codes:
0x0000 = Success
0x0396 = Invalid Date / Time data (Date / Time has not been set yet)

Example Request (Hex)

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

Example Response (Hex)

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

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

7.4.16 Extended Command 0x030E - Commit Configuration

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

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

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

Table 7-39 - Request Data for Extended Command 0x030E - Commit Configuration

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

Response Data: None

Result codes:
0x0000 = Success
0x0001 = Failure
0x0395 = Invalid Database Selector field

Example Request (Hex)

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

Example Response (Hex)

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

Complete Extended Data Length	0000
Extended Data	Not Applicable

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

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

The host uses this command to select a set of predetermined allowable values for the EMV configuration tags marked as only settable as part of Terminal Configuration in **Appendix C.2.1 EMV Contact Terminal Settings and Defaults (Contact Only)**. These values can not be set directly, they must be set to one of a specified set of values. Descriptions of the tags can be found in ***EMV Integrated Circuit Card Specifications for Payment Systems v4.3***. The host may set unrestricted tags directly using **Extended Command 0x0305 - Modify Terminal Configuration (MAC)** and **Extended Command 0x0307 - Modify Application Configuration (MAC)**.

(Fixed Key Only)
If the device is configured to use fixed key encryption using **Property 0x6B - Key Management Scheme (Fixed Key Only)** or the device's security level is less than 3, then MACing is not required. In this case, the Device Serial Number and MAC fields can be set all zeroes.

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

Table 7-40 - Request Data for Extended Command 0x0310 - Modify EMV Configuration (MAC, Contact Only)

Offset	Field Name	Value
0	Type of MAC	MAC algorithm designator 0x00 = ISO 9797 MAC Algorithm 3, Padding Method 1. (4 byte MAC)
1	Database Selector	0x00 = EMV Contact L2
2..17	Device Serial Number	16 Bytes DSN
18	Configuration Identifier	One byte field that specifies one of the following configurations identified in the device's Implementation Conformance Statement (ICS). Configuration Identifier = 0: (No MSR Fallback Only) • Attended, Online Only • SDA, DDA and CDA enabled • No MSR Fallback • Signature, No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 set to 21 • Tag 9F33 set to 20 28 C8 • Tag 9F40 set to 72 00 00 B0 01 Configuration Identifier = 1: (No MSR Fallback Only) • Attended, Online Only • SDA, DDA and CDA disabled • No MSR Fallback, Signature

		• No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 = 21 • Tag 9F33 = 20 28 00 • Tag 9F40 = 72 00 00 B0 01 Configuration Identifier = 2: (No MSR Fallback Only) • Attended, Offline/Online • SDA, DDA and CDA enabled • No MSR Fallback • Signature, No CVM Required • Goods, Services, Cashback, Payment • Print Attendant, Display Attendant/Cardholder, Code Table 1 • Tag 9F35 = 22 • Tag 9F33 = 20 28 C8 • Tag 9F40 = 72 00 00 B0 01
19..22	MAC	MAC computed on **Device Serial Number (DSN)** and **Configuration Identifier** fields. See section **7.4.1 About MACs**.

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

Result codes:
0x0000 = Success
0x0390 = Device Has No Keys
0x0391 = Invalid Device Serial Number
0x0392 = Invalid Type of MAC field
0x0393 = Invalid Slot Number field
0x0395 = Invalid Database Selector field
0x0397 = Invalid MAC
0x039C = Invalid Configuration Identifier

Example Request (Hex)

Header	
Command Number	49
Data Length	1D
Data	
Extended Data Offset	0000
Extended Command Number	0310
Complete Extended Data Length	0017
Extended Data	0000nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn01xxxxxxxx Where nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn is the 16-byte Device Serial Number (DSN) and xxxxxxxx is the 4-byte MAC

Example Response (Hex)

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

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

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

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

Table 7-41 - Request Data for Extended Command 0x0311 - Read EMV Configuration (Contact Only)

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

Table 7-42 - Response Data for Extended Command 0x0311 - Read EMV Configuration (Contact Only)

Offset	Field Name	Value
0	Configuration Identifier	One byte field that specifies one of the following configurations identified in the device's Implementation Conformance Statement (ICS). See the Configuration Identifier in **Extended Command 0x0310 - Modify EMV Configuration (MAC,** Contact Only) for a definition of possible values.

Result codes:
0x0000 = Success
0x0395 = Invalid Database Selector field

Example Request (Hex)

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

Example Response (Hex)

Header	
Result Code	0A
Data Length	07
Data	
Extended Data Offset	0000
Extended Result Code	0000

Complete Extended Data Length	0001
Extended Data	01

7.5 Command Group 0x04 - Auxiliary UART (Auxiliary Ports Only, Extended Commands Only) (mDynamo Gen I Only)

The host uses the commands in this section to control its auxiliary UART port, which enables customers to implement solutions where an external UART device, such as magnetic stripe reader or contactless reader, can communicate with the host by piggybacking on the device's connection to the host.

If the device is not connected to the host using UART, the device automatically converts between UART and the device/host connection type. This conversion may introduce data propagation delays and buffering limitations. MagTek advises thoroughly testing external devices with the auxiliary port to make sure they are completely compatible.

To configure the behavior of the auxiliary UART port, use **Property 0x69 - Auxiliary UART Configuration (Auxiliary Ports Only)**.

7.5.1 Extended Command 0x0400 - Auxiliary UART Transmit Data

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

The host uses this command to transmit data to the device's auxiliary UART. When the external UART device sends data to the device, the device passes the data to the host using **Notification 0x0400 - Auxiliary UART Received Data**.

Table 7-43 - Request Data for Extended Command 0x0400 - Auxiliary UART Transmit Data

Offset	Field Name	Value
0	Port Identifier	The identifier of the port to transmit on. Always set this field to zero.
1..n	Transmit Data	The data to transmit. If the length of the data to be transmitted exceeds 1023 bytes, the host must transmit it using multiple commands.

Response Data: None

Result codes:
0x0000 = Success
0x0481 = Port not opened

Example Request (Hex)

Header	
Command Number	49
Data Length	0A
Data	
Extended Data Offset	00 00
Extended Command Number	04 00
Complete Extended Data Length	00 04
Extended Data	00 31 32 33

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	00 00
Extended Result Code	00 00
Complete Extended Data Length	00 00
Extended Data	Not Applicable

7.5.2 Extended Command 0x0401 - Auxiliary UART Control

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

The host uses this command to control the device's auxiliary UART.

Table 7-44 - Request Data for Extended Command 0x0401 - Auxiliary UART Control

Offset	Field Name	Value
0	Port Identifier	The identifier of the port to transmit on. Always set this field to zero.
1	Control Data	The control data. If the host omits this field, the device returns the existing control data field to the host in the response data. See the Control Data Field table below for a description of each bit in this field.

Control Data Field: (Bit 0 is the least significant bit)

Bit	Field Name	Value
0	Open	0 = Port is closed. Ports power output is off, port signal lines will float, and data can not be transmitted or received on the port. 1 = Port is open.
1	DTR Level	0 = DTR output signal is set low when the port is open. 1 = DTR output signal is set high when the port is open.
2	RTS Level	0 = RTS output signal is set low when the port is open. 1 = RTS output signal is set high when the port is open.
3	DSR Level	0 = DSR input signal is read low when the port is open. 1 = DSR input signal is read high when the port is open. This bit always reads zero when the port is closed. This bit is read-only.
4	CTS Level	0 = CTS input signal is read low when the port is open. 1 = CTS input signal is read high when the port is open. This bit always reads zero when the port is closed. This bit is read only.
5..7	Reserved	These bits should always be written as zeroes.

Table 7-45 - Response Data for Extended Command 0x0401 - Auxiliary UART Control

Offset	Field Name	Value
0	Control Data	The control data. If the host transmitted Control Data in the request data, this field is not returned in the response data, otherwise the existing control data is returned in this field. See the Control Data Field table above for a description of each bit in this field.

Result codes:
0x0000 = Success

Example Request (Hex)

Header	
Command Number	49
Data Length	08
Data	
Extended Data Offset	00 00
Extended Command Number	04 01
Complete Extended Data Length	00 02
Extended Data	00 01

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	00 00
Extended Result Code	00 00
Complete Extended Data Length	00 00
Extended Data	Not Applicable

7.6 Command Group 0x05 - Auxiliary SPI (Auxiliary Ports Only, Extended Commands Only)

The host uses the commands in this section to control its auxiliary SPI port, which enables customers to implement solutions where an external SPI device, such as magnetic stripe reader or contactless reader, can communicate with the host by piggybacking on the device's connection to the host.

If the device is not connected to the host using SPI, the device automatically converts between SPI and the device/host connection type. This conversion may introduce data propagation delays and buffering limitations. MagTek advises thoroughly testing external devices with the auxiliary port to make sure they are completely compatible.

To configure the behavior of the auxiliary SPI port, use **Property 0x6A - Auxiliary SPI Configuration (Auxiliary Ports Only)**.

7.6.1 Extended Command 0x0500 - Auxiliary SPI Transmit and Receive Data

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

The host uses this command to transmit and receive data to communicate with an external SPI device.

The host calls this command when it needs to send data to the auxiliary SPI device, and should also call this command to read data from the auxiliary SPI port if it receives **Notification 0x0500 - Auxiliary SPI Data Change**.

Table 7-46 - Request Data for Extended Command 0x0500 - Auxiliary SPI Transmit and Receive Data

Offset	Field Name	Value
0	Port Identifier	The identifier of the port to transmit on. Always set this field to zero.
1..n	Transmit Data	The data to transmit. If the transmit data length exceeds 1023 bytes it must be sent using multiple commands.

Table 7-47 - Response Data for Extended Command 0x0500 - Auxiliary SPI Transmit and Receive Data

Offset	Field Name	Value
0..n	Receive Data	The data received. Since, for SPI, a byte is always received when a byte is transmitted, the number of bytes received is equal to the number of bytes transmitted.

Result codes:
0x0000 = Success
0x0581 = Port not opened
0x0582 = Data length too large

Example Request (Hex)

Header	
Command Number	49
Data Length	0A

Data	
Extended Data Offset	00 00
Extended Command Number	05 00
Complete Extended Data Length	00 04
Extended Data	00 FF FF FF

Example Response (Hex)

Header	
Result Code	0A
Data Length	09
Data	
Extended Data Offset	00 00
Extended Result Code	00 00
Complete Extended Data Length	00 03
Extended Data	FF FF FF

7.6.2 Extended Command 0x0501 - Auxiliary SPI Control

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

The host uses this command to control the device's auxiliary SPI.

Table 7-48 - Request Data for Extended Command 0x0501 - Auxiliary SPI Control

Offset	Field Name	Value
0	Port Identifier	The identifier of the port to transmit on. Always set this field to zero.
1	Control Data	The control data. If this field is omitted, the existing control data is returned in the response data. See the control data field table for a description of each bit in this field.

Control Data Field (Bit 0 is the least significant bit)

Bit	Field Name	Value
0	Open	If this bit is set to one, the port is open. When this bit is set to zero, the port is closed. When the port is closed, the port's power output is turned off, the port's signal lines will float, and data can not be transmitted or received on the port.
1	CS Level	If this bit is set to one, the CS (chip select) output signal is set high when the port is open. If this bit is set to zero, the CS output signal is set low when the port is open.
2	DAV Level	If this bit is set to one, the DAV (Data Available) input signal is read high when the port is open. If this bit is set to zero, the DAV input signal is read low when the port is open. This bit always reads zero when the port is closed. This bit is read only. Writing to this bit does nothing.
3-7	Reserved	These bits should always be written as zeroes.

Table 7-49 - Response Data for Extended Command 0x0501 - Auxiliary SPI Control

Offset	Field Name	Value
0	Control Data	The control data. If the control data field is included with the request data then this field is not returned in the response data, otherwise the existing control data is returned in this field. See the control data field table for a description of each bit in this field.

Result codes:
0x0000 = Success

Example Request (Hex)

Header	
Command Number	49
Data Length	08

Data	
Extended Data Offset	00 00
Extended Command Number	05 01
Complete Extended Data Length	00 02
Extended Data	00 01

Example Response (Hex)

Header	
Result Code	0A
Data Length	06
Data	
Extended Data Offset	00 00
Extended Result Code	00 00
Complete Extended Data Length	00 00
Extended Data	Not Applicable

8 Properties

8.1 About Properties

MagneSafe V5 devices have a number of programmable configuration properties stored in non-volatile memory. Most of the programmable properties pertain to data formats other than vendor-defined HID, but some of the properties deal with the device regardless of format (for information about changing formats and making format-specific properties visible, see **Property 0x10 - Interface Type**). These properties can be configured at the factory or by an administrator using software tools supplied by MagTek. Changing these configuration properties requires low-level communication with the device. Details for communicating with the device to read or change programmable properties are provided in section **7.3.1 Command 0x00 - Get Property** and section **7.3.2 Command 0x01 - Set Property (MAC)**.

8.2 Property 0x00 - Firmware ID

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

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

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	00

Example Get Response (Hex)

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

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

Property ID: `0x01`
Property Type: String
Length: 0 - 15 bytes
Get Property: Yes
Set Property: Yes
Default Value: Null string / ASCII device serial number set when the device is configured.

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

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	01

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	31 32 33

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

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

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

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

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	02

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	01

8.5 Property 0x03 - Device Serial Number

Property ID: 0x03
Property Type: String
Length: 0 - 15 bytes
Get Property: Yes
Set Property: Yes (Once)
Default Value: Null string / ASCII device serial number set when the device is configured.

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

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	03

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	31 32 33

8.6 Property 0x04 - MagneSafe Version Number

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

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

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	04

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	03	56 30 35

8.7 Property 0x07 - ISO Track Mask

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

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

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

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

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

Property ID: 0x0A
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes (Read-Only on some devices)
Default Value: 0x08 for all devices except eDynamo, kDynamo, mDynamo, DynaWave, and tDynamo, which use 0x40

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

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	0A

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	08

8.9 Property 0x10 - Interface Type

Property ID: 0x10
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes (No for devices that switch connections automatically)
Default Value: 0x00

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

- Valid values for this property are
  - 0x00 = USB HID (HID Only)
- On devices that have only one possible value for this property, the property is read-only.
- On devices that support multiple values for this property and do not handle connection switching automatically, the host can use this property to change the device's behavior. MagTek strongly recommends the host set this property before setting other properties, and immediately power cycle or reset the device (see **Command 0x02 - Reset Device)**, because it changes which other properties are available.

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	10

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	00

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

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

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

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

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

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	33

Example Get Response (Hex)

Result Code	Data Len	Property Value
00	01	01

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

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

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

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

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

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

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	52	02

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	52

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	02

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

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

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

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

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

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

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	02	67	01

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	67

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

8.13 Property 0x69 - Auxiliary UART Configuration (Auxiliary Ports Only)

Property ID: 0x69
Property Type: Word
Length: 2 bytes
Get Property: Yes
Set Property: Yes
Default Value: 0x0001 (9600 baud, port open)

This property controls the behavior of the auxiliary UART. For more information about the auxiliary UART, see section **7.5 Command Group 0x04 - Auxiliary UART (Auxiliary Ports Only, Extended Commands Only)**.

The property consists of a two byte word that contains bit fields that control various aspects of the auxiliary UART. This two byte property should be transmitted most significant byte first.

Auxiliary UART configuration: (Bit 0 is the least significant bit)

Bit	Field Name	Value
0	Open Initial State	1 = UART port opens automatically after a power cycle or reset.
1	Auto Close USB Suspend	1 = If the UART port is open when a USB suspend event occurs, the device automatically closes the UART port, and re-opens it upon USB resume. This bit can be used to help meet USB suspend current requirements for devices that draw power from the UART port.
2	(Contact Only) Auto Close Chip Card Power On	1 = If the UART port is open when a chip card power on event occurs, the device automatically closes the UART port, and re-opens it upon chip card power off. This bit can be used to avoid exceeding the amount of current the device can supply to both the chip card and the UART port at the same time for devices that draw power from the UART port.
3	DTR Initial Level	This bit specifies the initial level of the DTR signal after a power cycle or reset occurs. 0 = DTR output signal is set low when the port is open. 1 = DTR output signal is set high when the port is open.
4	RTS Initial State	This bit specifies the initial level of the RTS signal after a power cycle or reset occurs. 0 = RTS output signal is set low when the port is open. 1 = RTS output signal is set high when the port is open.
5..7	Reserved	These bits should always be written as zeroes.
8..10	Baud Rate	This bit field specifies the baud rate of the UART port. 0 = 9600 baud 1 = 19200 baud 2 = 38400 baud 3 = 57600 baud 4 = 115200 baud.
11	Reserved	This bit should always be written as zero.

12..13	Data Parity Stop	This bit field specifies the number of data bits, parity, and the number of stop bits for the UART port. 0 = 8N1 (8 data bits, no parity, 1 stop bit) 1 = 7E1 2 = 7O1 3 = 7M1
14..15	Reserved	These bits should always be written as zeroes.

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

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	03	69	00 01 (default)

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	69

Example Get Response (Hex)

Result Code	Data Len	Data
00	02	00 01 (default)

8.14 Property 0x6A - Auxiliary SPI Configuration (Auxiliary Ports Only)

Property ID: 0x6A
Property Type: Word
Length: 2 bytes
Get Property: Yes
Set Property: Yes
Default Value: 0x0311 (75000 baud, DAV notify high, port open)

This property controls the behavior of the auxiliary SPI. For more information about the auxiliary SPI, see section **7.6 Command Group 0x05 - Auxiliary SPI (Auxiliary Ports Only, Extended Commands** Only).

The property consists of a two byte word that contains bit fields that control various aspects of the auxiliary SPI. This two byte property should be transmitted most significant byte first.

Auxiliary SPI configuration (bit 0 is the least significant bit)

Bit	Field Name	Value
0	Open Initial State	1 = The SPI port opens automatically after a power cycle or reset.
1	Auto Close USB Suspend	1 = If the SPI port is open when a USB suspend event occurs, the device automatically closes the SPI port, and re-opens it upon USB resume This bit can be used to help meet USB suspend current requirements for devices that draw power from the SPI port.
2	CS Initial Level	This bit specifies the initial level of the CS (Chip Select) signal after a power cycle or reset occurs. 0 = CS output signal is set low when the port is open. 1 = CS output signal is set high when the port is open.
3	DAV Notify Low	1 = If the SPI port is open and the DAV (Data Available) input goes low, the device sends a **Notification 0x0500 - Auxiliary SPI Data Change** to the host.
4	DAV Notify High	1 = If the SPI port is open and the DAV (Data Available) input goes high, the device sends a **Notification 0x0500 - Auxiliary SPI Data Change** to the host.
5	Polarity	This bit specifies the idle state of the SPI clock signal. 0 = Clock signal low when idle. 1 = Clock signal high when idle (unsupported – always write as 0)
6	Phase	This bit specifies whether the SPI data signal is valid ***before*** or ***on*** the first SPI clock edge. 0 = SPI data signal available before the first SPI clock edge. 1 = SPI data signal available on the first SPI clock edge (unsupported – always write as 0)
7	Reserved	This bit should always be written as zero.

8..11	Baud Rate	This field specifies the baud rate of the SPI port. 0 = 9375 baud 1 = 18750 baud 2 = 37500 baud 3 = 75000 baud 4 = 150000 baud 5 = 300000 baud 6 = 600000 baud 7 = 1200000 baud 8 = 2400000 baud 9 = 4800000 baud.
12..15	Reserved	These bits should always be written as zeroes.

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

Example Set Request (Hex)

Cmd Num	Data Len	Property ID	Data
01	03	6A	03 11 (default)

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	6A

Example Get Response (Hex)

Result Code	Data Len	Data
00	02	03 11 (default)

8.15 Property 0x6B - Key Management Scheme (Fixed Key Only)

Property ID: 0x6B
Property Type: Byte
Length: 1 byte
Get Property: Yes
Set Property: Yes
Default Value: 0x00 (TDES DUKPT)

This property controls the key management scheme the device uses to generate keys to encrypt all data it sends to the host. When this property is set to Fixed Key, the host must use **Command 0x4E - Load Fixed Key (Fixed Key Only)** to load a 16-byte fixed key into the device. The device then uses the fixed key to encrypt all encrypted data it sends to the host. All references in this manual to the current DUKPT key should then be read to mean the single fixed key instead. For details about encryption, see section **5 Encryption, Decryption, and Key Management**.

The possible values are:

- 0x00 = **TDES DUKPT**. The device uses the current DUKPT Key to encrypt data.
- 0x01 = **Fixed Key**. The device uses the current fixed key as-is (no variant) to encrypt data.

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	6B

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	01

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

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

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

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

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

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

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

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

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

Example Set Request (Hex)

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

Example Set Response (Hex)

Result Code	Data Len	Data
00	00	

Example Get Request (Hex)

Cmd Num	Data Len	Property ID
00	01	6D

Example Get Response (Hex)

Result Code	Data Len	Data
00	01	02

Appendix A Examples

This section includes direct command examples and information about using demonstration software. In addition to the examples here, source code with detailed comments is included with the demo software and can be used as a guide for custom software development.

The book ***USB Complete*** by Jan Axelson is also a good guide for software developers, especially the chapter "Human Interface Devices: Host Applications."

A.1 Command Examples

This section provides examples of command sequences and cryptographic operations. Each example shows a sequence as it actually runs, so developers of custom software can check their code against the examples step-by-step to make sure the software is parsing and computing values correctly.

A.1.1 Example: Configuring a Device Before Encryption Is Enabled (HID Only)

This example configures the device to use the USB-HID data format (see section **2.1.3 How to Receive Data On the USB Connection**).

```
; This script demonstrates configuration commands for HID mode.
; It assumes the device is at Security Level 2 and that the Device
; Serial Number has never been set.
00 01 10      ; Get current interface
Request       : CMND=00, LEN=01, DATA=10
Response      : RC=  00, LEN=01, DATA=01

01 02 10 00   ; Set Interface to HID
Request       : CMND=01, LEN=02, DATA=10 00
Response      : RC=  00, LEN=00, DATA=

02 00         ; Reset so changes take effect
Request       : CMND=02, LEN=00, DATA=
Response      : RC=  00, LEN=00, DATA=

Delay         : (waited 5 seconds)

00 01 10      ; Get current interface (should return 0)
Request       : CMND=00, LEN=01, DATA=10
Response      : RC=  00, LEN=01, DATA=00
```

A.1.2 Example: Changing from Security Level 2 to Security Level 3

```
; This script demonstrates changing from Security Level 2 to Security
Level 3.
; It assumes the device is at Security Level 2 with the ANS X9.24
Example
; key loaded and the KSN counter set to 1.
09 00         ; Get current KSN (should be FFFF9876543210E00001)
Request       : CMND=09, LEN=00, DATA=
Response      : RC=  00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 01

; For KSN 1, MAC Key: 042666B4918430A3 68DE9628D03984C9
;
; The command to change Security Level looks like: 15 05 03 nnnnnnnn
;  where nnnnnnnn is the MAC.
;
; The data to be MACd is: 15 05 03
; Data to be MACd must be in blocks of eight bytes, so we left justify
and
; zero fill the block to get: 15 05 03 00 00 00 00 00 (This is the
block to MAC)
; For convenience show it as the compacted form: 1505030000000000
;
; The MAC algorithm run with this data uses the following
cryptographic
; operations:
;
;  Single DES Encrypt the data to be MACd with the left half of the
MAC Key:
;     1505030000000000 1DES Enc with 042666B4918430A3 =
BFBA7AE4C1597E3D
;
;  Single DES Decrypt the result with the right half of the MAC Key:
;     BFBA7AE4C1597E3D 1DES Dec with 68DE9628D03984C9 =
DA91AB9A8AD9AB4C
;
;  Single DES Encrypt the result with the left half of the MAC Key:
;     DA91AB9A8AD9AB4C 1DES Enc with 042666B4918430A3 =
E7E2FA3882BB386C
;
; The leftmost four bytes of the final result are the MAC = E7E2FA38
;
; Send the MACd Set Security Level command
15 05 03 E7E2FA38
Request       : CMND=15, LEN=05, DATA=03 E7 E2 FA 38
Response      : RC=  00, LEN=00, DATA=

02 00         ; Reset so changes take effect
Request       : CMND=02, LEN=00, DATA=
Response      : RC=  00, LEN=00, DATA=

Delay         : (waited 5 seconds)
```

```
09 00         ; Get current KSN (should be FFFF9876543210E00002)
Request       : CMND=09, LEN=00, DATA=
Response      : RC=  00, LEN=0A, DATA=FF FF 98 76 54 32 10 E0 00 02

15 00         ; Get current Security Level (Should be 03)
Request       : CMND=15, LEN=00, DATA=
Response      : RC=  00, LEN=01, DATA=03
```

A.2 About the SDKs and Additional Examples

MagTek provides SDKs and corresponding documentation for many programming languages and operating systems that enable software developers to quickly develop custom host software that communicates with this device, without having to deal with the complexities of platform APIs for direct communication across the various available connection types, connecting using the various available communication protocols, and parsing the various available data formats.

The SDKs and corresponding documentation include:

- Functions for sending the direct commands described in this manual
- Wrappers for commonly used commands and properties that further simplify development
- Detailed compilable examples of processing incoming payment data and using the direct commands and properties described in this manual

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

Appendix B EMV Message Formats (EMV Only)

B.1 ARQC Messages (EMV Only)

This section gives the format of the ARQC Message delivered in **Notification 0x0303 - ARQC Message**. The contents of the ARQC Message is slightly different depending on whether the device is set to **Security Level 2** (not encrypting) or **Security Level 3** (encrypting). Support for EMV transactions at **Security Level 2** is only available on mDynamo.

B.1.1 ARQC Message Format Security Level 2

When the device is set to **Security Level 2** (not encrypting), the ARQC Message TLV data object contains the following:

```
F9<len> /* container for MAC structure and generic data */
          DFDF54(MAC KSN)<len><val>
          DFDF55(MAC Encryption Type)<len><val>
          DFDF25(IFD Serial Number)<len><val>
          FA<len>/* container for generic data */
               70<len> /*container for ARQC */
                    DFDF53<len><value> /*fallback indicator */
                    5F20<len><value> /*cardholder name */
                    5F30<len><value> /*service code */
                    DFDF4D<len><value> /* Masked T2 PICC/ICC Data */
                    DFDF52<len><value> /* card type */
                    <tags defined by DFDF02 >

(Buffer if any to be a multiple of 8 bytes)

CBC-MAC (4 bytes reserved, not calculated)
```

If the device is configured to prefer MSD data, it includes that data in additional TLV data objects in TLV data object 70.

The device populates TLV data object DFDF53 with one of the following fallback indicators:

- 0x00 = No fallback or missing tag
- 0x01 = Technical Fallback used
- 0x81 = MSR Fallback used

The device populates TLV data object DFDF52 with one of the following card types:

- 0x00 = Other
- 0x01 = Financial
- 0x02 = AAMVA
- 0x03 = Manual
- 0x04 = Unknown
- 0x05 = ICC
- 0x06 = Contactless ICC - EMV
- 0x07 = Financial MSR and ICC
- 0x08 = Contactless ICC - MSD

The device constructs the contents of tag DFDF4D, using EMV transaction data to emulate track 2 data as though it came from an ISO/ABA magnetic stripe card. Much of the data is masked; the device sends a specified mask character instead of the actual character from the transaction. The device provides masking settings in **Property 0x07 - ISO Track Mask**, which allows the host software to specify masking details for the Primary Account Number, the masking character to be used, and whether a correction should be applied to make the Mod 10 (Luhn algorithm) digit at the end of the PAN be correct.

Table 8-1 provides an example of track 2 data as it would appear if the device sent it in the clear. **Table 8-2** shows the same data as it might appear with a specific set of masking rules applied.

Table 8-1 – Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted

Sample ISO/ABA Swiped Track Data, Clear Text / Decrypted	
Track 2	;6011000995500000=15121015432112345678?

Table 8-2 – Sample ISO/ABA Swiped Track Data, Masked

Sample ISO/ABA Swiped Track Data, Masked	
Track 2	;6011000020000000=15120000000000000000?

Table 8-3 shows an example of track 2 data using unmasked placeholders to make it easier to see the relative positions of the values embedded in the track data, and can be interpreted as follows:

- ? and ; are Sentinels / delimiters.
- The string of 5s is the Account Number / PAN.
- The string of 3s is the Expiration Date.
- The string of 8s is the Service Code.
- The remaining characters (0s, 4s, and 6) are Discretionary Data, which is of varying length and content and comes from the card, and must be interpreted according to the standards established by issuers, payment brands, and so on.

Table 8-3 – Example Generic ISO/ABA Track Data Format

Generic ISO/ABA Track Data Format	
Track 2 Data	;5555555555555555=33338880004444006?

The device masks the data as follows:

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

- All other characters are set to the specified mask character.

B.1.2 ARQC Message Format Security Level 3

When the device is set to **Security Level 3** (encrypting), the ARQC Message TLV data object contains the following:

```
F9<len> /* container for MAC structure and generic data */
          DFDF54(MAC KSN)<len><val>
          DFDF55(MAC Encryption Type)<len><val>
          DFDF25(IFD Serial Number)<len><val>
          FA<len>/* container for generic data */
               70<len> /*container for ARQC */
                    DFDF53<len><value> /*fallback indicator */
                    5F20<len><value> /*cardholder name */
                    5F30<len><value> /*service code */
                    DFDF4D<len><value> /* Masked T2 PICC/ICC Data */
                    DFDF52<len><value> /* card type */
                    F8<len> /*container tag for encryption */
                         DFDF59(Encrypted Data
Primitive)<len><Encrypted Data val (Decrypt data to read tags)>
                         DFDF56(Encrypted Transaction Data
KSN)<len><val>
                         DFDF57(Encrypted Transaction Data
Encryption Type)<val>
                         DFDF58(# of bytes of padding in
DFDF59)<len><val>

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The values inside tags DFDF52, DFDF53, and DFDF4D are fully described in section **B.1.1**.

The device encrypts the Value inside data container DFDF59 using the **Data Encryption, request or both ways** variant [or other variant depending on **Property 0x67 - EMV Data Encryption Variant (EMV Only)**] of the current DUKPT Key used in the relevant transaction. As a requirement for using the DUKPT TDES encryption algorithm, the device pads it so the length of its value is a multiple of 8 bytes. The device uses tag DFDF58 to report how many bytes of tag DFDF59 are padding. After the host decrypts it, DFDF59 contains a list of TLV data objects defined by terminal setting DFDF02 or DFDF08 is card type is contactless-MSD. For example:

```
                    FC<len>/* container for encrypted generic data *
                         <tags defined by DFDF02 or DFDF08>
                         F4<len>/* container tag for encrypted MSR
data */
                              DFDF36 <EncT1status><len><val>
                              DFDF37 <EncT1data><len><val>
                              DFDF38 <EncT2status><len><val>
                              DFDF39 <EncT2data><len><val>
                              DFDF3A <EncT3status><len><val>
                              DFDF3B <EncT3data><len><val>
                              DFDF3C <Encrypted Magneprint
Data><len><val>
```

```
                                    DFDF43 <Magneprint Status
Data><len><val>
                                    DFDF50(MSR KSN Data)<len><val> /*sent
in the clear*/
                                    DFDF51(MSR EncryptionType)<len><val>

                        <Padding to force DFDF59 plus padding to be a
multiple of 8 bytes>
```

If the device is configured to prefer MSD data, it includes that data in additional TLV data objects in TLV data object FC.

B.2 ARPC Response from Online Processing (EMV Only)

This section specifies the format of the data for **Extended Command 0x0303 - Online Processing Result / Acquirer Response (EMV Only)**. The host sends this request to the device in response to **Notification 0x0303 - ARQC Message**.

An ARPC Response is a TLV data object with the following contents:

```
F9<len> /* container for MAC structure and generic data */
     DFDF54 (MAC KSN)<len><val>
     DFDF55 (Mac Encryption Type><len><val>
     DFDF25 (IFD Serial Number)<len><val>
FA<len> /* Container for generic data */
     70 04 8A 02 30 30
     (ARPC padding, if any, to be a multiple of 8 bytes)
CBC-MAC (4 bytes, reserved, must be sent to the device, however, the
device does not check for the properly calculated CBC-MAC)
```

B.3 Transaction Result Messages (EMV Only)

This section specifies the format for data the device sends using **Notification 0x0304 - Transaction Result Message**. The format is controlled by **Property 0x68 – EMV Message Format**.

TLV data object `DFDF1A` contains one of the following Transaction Status values:

- 0x00 = Approved
- 0x01 = Declined
- 0x02 = Error
- 0x10 = Cancelled by Host
- 0x1E = Manual Selection Cancelled by Host
- 0x1F = Manual Selection Timeout
- 0x21 = Waiting for Card Cancelled by Host
- 0x22 = Waiting for Card Timeout
- 0x23 = Reserved
- 0xFF = Unknown

The format of Transaction Result messages depends on whether the device is set to **Security Level 2** (not encrypting) or **Security Level 3** (encrypting). Support for EMV transactions at **Security Level 2** is only available on mDynamo.

B.3.1 Transaction Result Message Format Security Level 2

When the device is set to **Security Level 2** (not encrypting), the Transaction Result TLV data object contains the following:

```
F9<len> /* container for MAC structure and generic data */
     DFDF54(MAC KSN)<len><val>
     DFDF55(MAC Encryption Type)<len><val>
     DFDF25(IFD Serial Number)<len><val>
     FA<len>/* container for generic data */
          F0<len> /* Transaction Results */
               F1<len> /* container for Status Data */
                    /* Status Data tags */
                    DFDF1A - Transaction Status
                    DFDF1B - Additional Transaction Information

               F2<len>/* container for Transaction Data */
                    /* Data tags (defined by DFDF17) */

               F3<len>/* container for Reversal Data, if any */
                    /* Reversal Data tags (defined by DFDF05)*/

               F7<len>/* container for Merchant Data */
                    /* Merchant Data tags */
                    5F25<len> /* Application Effective Date */
                    5F24<len> /* Application Expiration Date */
                    89<len> /* Authorization Code */
                    5F2A<len> /* Transaction Currency Code */
                    9F02<len> /* Amount, Authorized */
                    9F03<len> /* Amount, Other */
                    9F06<len> /* Application Identifier */
                    9F12<len> /* Application Preferred Name */
                    9F1C<len> /* Terminal Identification */
                    9F39<len> /* POS Entry Mode */
                    9C<len> /* Transaction Type */
                    9F34<len> /* Cardholder Verification Results */
                    5F57<len> /* Account Type */
                    5F20<len> /* Cardholder Name */
                    DFDF4D<len> /* Masked T2 PICC/ICC Data */

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The value inside tag DFDF4D is fully described in section **B.1.1**.

If the device is configured to prefer MSD data, it includes that data in additional TLV data objects in TLV data object F2.

B.3.2 Transaction Result Message Format Security Level 3

When the device is set to **Security Level 3** (encrypting), the Transaction Result TLV data object contains the following:

```
F9<len> /* container for MAC structure and generic data */
      DFDF54(MAC KSN)<len><val>
      DFDF55(MAC Encryption Type)<len><val>
      DFDF25(IFD Serial Number)<len><val>
      FA<len>/* container for generic data */
            F0<len> /* Transaction Results */

                  F1<len> /* container for Status Data */
                        … /* Status Data tags */
                        DFDF1A - Transaction Status
                        DFDF1B - Additional Transaction Information

                  F8<len> /* container tag for encryption */
                        DFDF59(Encrypted Data 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 */
                        5F25<len> /* Application Effective Date */
                        5F24<len> /* Application Expiration Date */
                        89<len> /* Authorization Code */
                        5F2A<len> /* Transaction Currency Code */
                        9F02<len> /* Amount, Authorized */
                        9F03<len> /* Amount, Other */
                        9F06<len> /* Application Identifier */
                        9F12<len> /* Application Preferred Name */
                        9F1C<len> /* Terminal Identification */
                        9F39<len> /* POS Entry Mode */
                        9C<len> /* Transaction Type */
                        9F34<len> /* Cardholder Verification Results */
                        5F57<len> /* Account Type */
                        5F20<len> /* Cardholder Name */
                        DFDF4D<len> /* Masked T2 PICC/ICC Data */

(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes reserved, not calculated)
```

The value inside tag DFDF4D is fully described in section **B.1.1**.

The device encrypts the Value inside data container DFDF59 using the **Data Encryption, request or both ways** variant [or other variant depending on **Property 0x67 - EMV Data Encryption Variant (EMV Only)**] of the current DUKPT Key used in the relevant transaction. As a requirement for using the DUKPT TDES encryption algorithm, the device pads it so the length of its value is a multiple of 8 bytes. The device uses tag DFDF58 to report how many bytes of tag DFDF59 are padding. After the host

decrypts it, DFDF59 contains a list of TLV data objects defined by terminal setting DFDF17 and DFDF05. For example:

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

If the device is configured to prefer MSD data, it includes that data in additional TLV data objects in TLV data object F2.

Appendix C EMV Terminal and Application Settings (EMV Only)

C.1 EMV Common Settings

This section lists settings that are common across all EMV databases on the device.

C.1.1 EMV Common Terminal Settings and Defaults

This section lists the default EMV Terminal Settings shared across all terminal databases on the device. For information about reading and changing these settings, see section **7.4.8 Extended Command 0x0306 - Read Terminal Configuration** and section **7.4.7 Extended Command 0x0305 - Modify Terminal Configuration**.

Table 8-4 - EMV Common Terminal Settings

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
5F2A	08 40	0x02	MagTek	Transaction Currency Code. Valid values are the numerical codes from ***ISO 4217 Codes for the representation of currencies***, for example: • 0x0000 = Use Selected Application's Currency Code Terminal Setting • 0x0840 = US Dollar • 0x0978 = Euro
5F36	02	0x01	MagTek	Transaction Currency Exponent
9F1A	08 40	0x02	MagTek	Terminal Country Code. The device's terminal country codes are numeric and derived from ***ISO 3166-1***, for example: • 0840 = United States • 0250 = France • 0380 = Italy • 0724 = Spain • 0276 = Germany
9F1C	31 31 32 32 33 33 34 34	0x08	MagTek	Terminal Identification
9F4E	30 30 30 30 30 30 30	0x28	MagTek	Merchant Name and Location
DFDF14	00 00 75 30	0x04	MagTek	Socket Timeout for Online Processing (ms)
DFDF15	00 00 00 01	0x04	MagTek	Socket Retries (number of connection retries in Online Processing)
DFDF19	65 6E	0x02	MagTek	Reserved
DFDF2D	65 6E 64 65	0x0A	Read Only	Reserved

C.1.2 EMV Common Application Settings and Defaults

There are no default EMV Application Settings shared across all application databases on the device.

C.2 EMV Contact Settings (Contact Only)

C.2.1 EMV Contact Terminal Settings and Defaults (Contact Only)

This section lists the default EMV Contact Terminal default settings. For information about reading and changing these settings, see section **7.4.8 Extended Command 0x0306 - Read Terminal Configuration** and section **7.4.7 Extended Command 0x0305 - Modify Terminal Configuration**.

Table 8-5 - EMV Contact Terminal Settings

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Terminal Settings** from section **C.1.1**				
9F15	30 30	0x02	MagTek	Merchant Category Code
9F16	30 30 30 30 30 30 30	0x0F	MagTek	Merchant Identifier
9F33	20 28 C8	0x03	MagTek	Terminal Capabilities (Set by Terminal Configuration, see section **7.4.17**)
9F35	21	0x01	MagTek	Terminal Type (Set by Terminal Configuration, see section **7.4.17**)
9F3C	09 98	0x02	MagTek	Transaction Reference Code
9F3D	02	0x01	MagTek	Transaction Currency Exponent
9F40	72 00 00 B0 01	0x05	MagTek	Additional Terminal Capabilities (Set by Terminal Configuration, see section **7.4.17**)
DFDF01	A0 00 00 00 04 F8 00 10 00	0x09	MagTek	Certificate Revocation List
DFDF02	9A DF DF 28 9F 02 5A 89 9F 10 9F 15 9F 16 9F 4E 82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 26 9F 27 9F 36 9C 9F 33 9F 34 9F 37 9F 39 9F 40 95 9B 9F 5B DF DF 00 9F 1E 9F 1A 5F 2A 9F 01 9F 21 8A DF 81 20 DF 81 21 DF 81 22 5F 20 50 5F 34 84 9F 03 9F 09 9F 1E 9F 35 9F 41 9F 53 F4	0x81	MagTek	Online message for EMV transaction

DFDF05	9A 82 9F 36 9F 1E 9F 10 9F 5B 9F 33 9F 35 95 9F 01 5F 24 5A 5F 34 8A 9F 15 9F 16 9F 39 9F 1A 9F 1C 57 9F 02 5F 2A 9F 21 9C	0x80	MagTek	Reversal message for EMV transaction
DFDF06	8A 91	0x02	MagTek	Tags participating in online response
DFDF16	00 00 00 80	0x04	MagTek	Maximum length of issuer script (Read Only)
DFDF17	9A DF DF 28 9F 02 9F 03 5A 89 9F 10 9F 15 9F 16 9F 4E 82 8E 5F 24 5F 25 9F 06 9F 07 9F 0D 9F 0E 9F 0F 9F 26 9F 27 9C 9F 33 9F 34 9F 35 9F 36 9F 37 9F 39 9F 40 9F 41 9F 53 95 9B 9F 5B DF DF 00 9F 1E 9F 1A 5F 2A 9F 01 8A DF 81 20 DF 81 21 DF 81 22 5F 20 5F 34 9F 09 84	0x80	MagTek	EMV Transaction Result Message Tags

DFDF20	43 28	0x02	Read Only	Terminal Features, read only Byte 1: Bit 8 TAC/IAC-Default process when unable to go online Bit 7 Manual Language Selection Enabled Bit 6 Referrals are supported Bit 5 CDA Failure detected prior to TAA is enabled Bit 4/Bit 3 0b00 = CDA Mode 1 is enabled, 0b01 = CDA Mode 2 is enabled, 0b10 = CDA Mode 3 is enabled, 0b11 = CDA Mode 4 is enabled Bit 2 Cardholder Confirmation is enabled Bit 1 EMV Language Selection is enabled Byte 2: Bit 8 RFU Bit 7 'Forced Acceptance' is enabled Bit 6 'Application Preferred Order' is enabled Bit 5 'Transaction log' is enabled Bit 4 'Revocation of Issuer Public Key' is enabled Bit 3 'Account Type selection' is enabled Bit 2 'Subsequent Bypass PIN Entry' is enabled Bit 1 'Bypass PIN Entry' is enabled
DFDF26	4D 41 47 54 45 4B 20 44 45 46 41 55 4C 54	0x10	MagTek	EMV Database Label
DFDF5B	0C	0x01	MagTek	Terminal Capabilities for Purchase transaction
DFDF5C	02	0x01	MagTek	Terminal Capabilities for Cashback transaction
DFDF67	01	0x01	MagTek	Reserved
DFDF6E	0C	0x01	MagTek	Terminal Capabilities for Payment transaction
DFDF75	0C	0x01	MagTek	Terminal Capabilities for Inquiry (Not Supported)
DFDF76	0C	0x01	MagTek	Terminal Capabilities for Transfer (Not Supported)

C.2.2 EMV Contact Application Settings and Defaults (Contact Only)

This section lists the default EMV Contact Application Settings. For information about reading and changing these settings, see section **7.4.10 Extended Command 0x0308 - Read Application Configuration** and section **7.4.9 Extended Command 0x0307 - Modify Application Configuration**.

Table 8-6 - EMV Contact Application Slot 1 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 25 01	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 25 01	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-7 - EMV Contact Application Slot 2 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 06 20 06 20	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 06 20 06 20	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	00 00 00	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 AC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-8 - EMV Contact Application Slot 3 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 01 52 30 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 01 52 30 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 27 10	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 00 00 20 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC E0 9C F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-9 - EMV Contact Application Slot 4 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 98 08 40	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 98 08 40	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 00 00 20 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC E0 9C F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-10 - EMV Contact Application Slot 5 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 02 77 10 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 02 77 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 01	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 F8 A8 F0	0x05	MagTek	Terminal Action Code - Default
DF8121	10 10 58 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC F8 E4 B8 70	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-11 - EMV Contact Application Slot 6 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 65 10 10	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 65 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	02 00	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 60 24 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 60 AC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-12 - EMV Contact Application Slot 7 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 04 10 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 B8 A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 B8 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-13 - EMV Contact Application Slot 8 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 04 30 60	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 30 60	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 BC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-14 - EMV Contact Application Slot 9 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 04 22 03	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06 5F 2A 02 9A 03 9C 01 95 05 9F 37 04	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 04 22 03	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 02	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	FC 50 BC A0 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 00 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	FC 50 BC F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-15 - EMV Contact Application Slot 10 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 03 33 01 01 01	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 01	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-16 - EMV Contact Application Slot 11 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 03 33 01 01 02	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 02	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-17 - EMV Contact Application Slot 12 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 03 33 01 01 03	0x10	MagTek	Dedicated File (DF) Name
97	00 00 00	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 03 33 01 01 03	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 20	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	D8 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-18 - EMV Contact Application Slot 13 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 03 10 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 10 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-19 - EMV Contact Application Slot 14 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 03 20 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 20 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-20 - EMV Contact Application Slot 15 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	A0 00 00 00 03 30 10	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	A0 00 00 00 03 30 10	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Table 8-21 - EMV Contact Application Slot 16 Data

Tag	Default Value (Hex)	Max. Length	Configurable	Tag Description
All **EMV Common Application Settings and Defaults** from section **C.1.2**				
84	00	0x10	MagTek	Dedicated File (DF) Name
97	9F 02 06	0x0F	MagTek	Default TDOL
9F01	00 00 00 00 00 01	0x06	MagTek	Acquirer Identifier
9F06	00	0x10	MagTek	Application Identifier (AID) - terminal
9F09	00 8C	0x02	MagTek	Application Version Number
9F1B	00 00 00 00	0x04	MagTek	Terminal Floor Limit
9F49	9F 37 04	0x0A	MagTek	Default DDOL
DFDF23	01	0x01	MagTek	Application Selection Indicator (ASI)
DF8120	DC 40 00 A8 00	0x05	MagTek	Terminal Action Code - Default
DF8121	00 10 00 00 00	0x05	MagTek	Terminal Action Code - Denial
DF8122	D8 40 04 F8 00	0x05	MagTek	Terminal Action Code - Online
DFDF10	00 00 00 00 00 00	0x06	MagTek	Threshold Value for Biased Random Selection
DFDF11	63	0x01	MagTek	Target Percentage to be used for Random Selection (0 - 63 hex)
DFDF12	63	0x01	MagTek	Maximum Target Percentage to be used for Biased Random Selection (0 - 63 hex)
DFDF68	00	0x01	MagTek	PIN Bypass Supported (Not Supported)

Appendix D Warranty, Standards, and Certifications

LIMITED WARRANTY

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

During the Warranty Period, should this product fail to conform to MagTek's specifications, MagTek will, at its option, repair or replace this product at no additional charge except as set forth below. Repair parts and replacement products will be furnished on an exchange basis and will be either reconditioned or new. All replaced parts and products become the property of MagTek. This limited warranty does not include service to repair damage to the product resulting from accident, disaster, unreasonable use, misuse, abuse, negligence, or modification of the product not authorized by MagTek. MagTek reserves the right to examine the alleged defective goods to determine whether the warranty is applicable.

Without limiting the generality of the foregoing, MagTek specifically disclaims any liability or warranty for goods resold in other than MagTek's original packages, and for goods modified, altered, or treated without authorization by MagTek.

Service may be obtained by delivering the product during the warranty period to MagTek (1710 Apollo Court, Seal Beach, CA 90740). If this product is delivered by mail or by an equivalent shipping carrier, the customer agrees to insure the product or assume the risk of loss or damage in transit, to prepay shipping charges to the warranty service location, and to use the original shipping container or equivalent. MagTek will return the product, prepaid, via a three (3) day shipping service. A Return Material Authorization ("RMA") number must accompany all returns. Buyers may obtain an RMA number by contacting MagTek Support Services at (562) 546-6800.

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 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: Changes or modifications not expressly approved by MagTek could void the user's authority to operate this equipment.

CANADIAN DECLARATION OF CONFORMITY

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.

INDUSTRY CANADA (IC) RSS

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) This device may not cause interference, and (2) This device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) L'appareil ne doit pas produire de brouillage, et (2) L'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

CUR/UR

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

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.

EU STATEMENT

Hereby, MagTek Inc. declares that the radio equipment types **Wideband Transmission System** (802.11 wireless and Bluetooth Low Energy), and **Non-Specific Short Range Device** (contactless) are in compliance with ***Directive 2014/53/EU***. The full text of the EU declarations of conformity is available at the following internet addresses:

- https://www.magtek.com/Content/DocumentationFiles/D998200238.pdf.
- https://www.magtek.com/Content/DocumentationFiles/D998200296.pdf

AUSTRALIA / NEW ZEALAND STATEMENT

Testing for compliance with AS/NZS standards was performed by a registered and accredited laboratory. The unit under test was found compliant with standards established under AS/NZS CISPR 32 (2013), AS/NZS 4268 Table 1, Row 59 DTS 2400-2483MHz SRD (802.11), and AS/NZS 4268 (2017) Table 1, Row 43 13.553-13.567MHz (contactless reader).

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) Directive (EU) 2015/863 amending Annex II to Directive 2011/65/EU. The marking is clearly recognizable, either as written words like “Pb-free,” “lead-free,” or as another clear symbol (Pb).

PCI STATEMENT

PCI Security Standards Council, LLC (“PCI SSC”) has approved this PIN Transaction Security Device to be in compliance with PCI SSC’s PIN Security Requirements.

When granted, PCI SSC approval is provided by PCI SSC to ensure certain security and operational characteristics important to the achievement of PCI SSC’s goals, but PCI SSC approval does not under any circumstances include any endorsement or warranty regarding the functionality, quality or performance of any particular product or service. PCI SSC does not warrant any products or services provided by third parties. PCI SSC approval does not under any circumstances include or imply any product warranties from PCI SSC, including, without limitation, any implied warranties of merchantability, fitness for purpose, or non-infringement, all of which are expressly disclaimed by PCI SSC. All rights and remedies regarding products and services which have received PCI SSC approval shall be provided by the party providing such products or services, and not by PCI SSC.

SAFETY

This product has been evaluated by multiple safety certification agencies, including Underwriters Laboratories (UL) and the United States Federal Communications Commission (FCC Class A and Class B), and is designed to protect both the user and the device. This document is written specifically to work in conjunction with these safety and integrity features to protect the user and the device. It is very important to follow all steps in the product documentation carefully, in the order in which they are described, and at the recommended times. Failure to do so could result in personal injury, and / or cause damage to the device, and / or void the product warranty.

SAFETY REQUIREMENTS

⚠ CAUTION

Never do any of the following:

- DO NOT use a ground adapter plug to connect equipment to a power socket-outlet that lacks a ground connection terminal.
- DO NOT attempt any maintenance function that is not specifically described in this manual or in other ExpressCard 3000 instructional documents published by MagTek.
- DO NOT remove any of the covers or guards that are fastened with screws. There are no user-serviceable areas within these covers.
- DO NOT override or "cheat" electrical or mechanical interlock devices.
- DO NOT use EC3000 supplies or cleaning materials for other than their intended purposes.
- DO NOT operate the equipment if you or anyone else have noticed unusual noises or odors.

Consider the following before operating the ExpressCard 3000:

- Connect the EC3000 to a properly grounded AC power socket-outlet. If in doubt, have the socket-outlet checked by a qualified electrician. Improper connection of the device's grounding conductor creates a risk of electric shock.
- Place the EC3000 on a solid surface that can safely support the device's weight plus the weight of a person leaning against it (such as a service technician).
- Be careful when moving or relocating the device. Use proper lifting techniques.
- Use materials and supplies specifically designed for MagTek devices. Using unsuitable materials may result in poor performance, and in some cases may be hazardous.

SOFTWARE LICENSE AGREEMENT

IMPORTANT: YOU SHOULD CAREFULLY READ ALL THE TERMS, CONDITIONS AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE INSTALLING THE SOFTWARE PACKAGE. YOUR INSTALLATION OF THE SOFTWARE PACKAGE PRESUMES YOUR ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE PACKAGE AND ASSOCIATED DOCUMENTATION TO THE ADDRESS IN THIS DOCUMENT, ATTENTION: CUSTOMER SUPPORT.

TERMS, CONDITIONS, AND RESTRICTIONS

MagTek, Incorporated (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software."

LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with MagTek products. LICENSEE MAY NOT COPY, MODIFY, OR TRANSFER THE SOFTWARE IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt to reverse engineer the Software. Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software or license to the Software to another party without the prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted. Licensee may not copy the Software except for archival purposes or to load for execution purposes. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded are free from defects in material or workmanship under normal use.

THE SOFTWARE IS PROVIDED AS IS. LICENSOR MAKES NO OTHER WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Because of the diversity of conditions and hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s) if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void, or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the laws of the State of California and shall inure to the benefit of MagTek, Incorporated, its successors or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS, AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL VERBAL AND WRITTEN COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO MAGTEK, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE ADDRESS LISTED IN THIS DOCUMENT, OR E-MAILED TO SUPPORT@MAGTEK.COM.

DEMO SOFTWARE / SAMPLE CODE: Unless otherwise stated, all demo software and sample code are to be used by Licensee for demonstration purposes only and MAY NOT BE incorporated into any production or live environment. The PIN Pad sample implementation is for software PIN Pad test purposes only and is not PCI compliant. To meet PCI compliance in production or live environments, a third-party PCI compliant component (hardware or software-based) must be used.