MAGTEK®
SECURITY FROM THE INSIDE

SDK - MagneFlex Powder, MIddleware,

PIN PEDs

Programmer's Manual (MagneFlex API)

December 2021

Manual Part Number:
D998200118-80

REGISTERED TO ISO 9001:2015

Table 0.1 – Revisions

Rev Number	Date	Notes
10	03/02/2016	Initial Release
20	03/22/2016	Renamed SendAPRCRequest to RequestSendAcquirerResponse. Added RequestOperationStatus.
30	07/15/2016	Added “Request” to page 7. Added “03= Read all tags operation” to RequestEMVTags. Deleted “Tones” and “FiedSeparator” from RequestManualSwipe. Added “AdditionalRequestData” to RequestSmartCard Updated Code in Appendix A.1, A.2, A.3.
40	10/02/2017	Changed the parameter Timeout to WaitTime in the RequestSendAcquirerResponse operation. Added details for the Reserved bytes of the RequestSmartCard operation.
50	11/02/2017	Added WaitTimeBeforeTransactionComplete to the RequestSendAcquirerResponse operation
60	05/09/2018	Added the CloseDevice and EndSession parameters to requests.
70	02/18/2021	Added the URI for BLE device connection type.
80	12/10/2021	Added support for QwickChipMode. Added RequestSmartCardEx, ReleaseDeviceEx, and RequestDeviceStatus.

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 ON THE FRONT PAGE OF 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 PC hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

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

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

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

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO MAGTEK, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE 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.

Table of Contents

1 Introduction

This document provides instructions for software developers who want to create software solutions that include a IPAD, DynaPro, and DynaPro Mini connected to a Windows-based host via USB, BLE, or by Ethernet. It is part of a larger library of documents designed to assist IPAD, DynaPro, and DynaPro Mini implementers, which includes the following documents available from MagTek:

- ***D99875585 DYNAPRO PROGRAMMER'S MANUAL (COMMANDS)***
- ***D99875629 DYNAPRO MINI PROGRAMMER'S MANUAL (COMMANDS)***
- ***D99875430 IPAD PROGRAMMER'S MANUAL (COMMANDS)***

1.1 Nomenclature

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 meanings in the context of USB communication than it does in the context of networked financial transaction processing. In this document, "device" and "host" are used strictly as follows:

- **Device** refers to the MSR device (eg. Dynamag) that receives and responds to the command set specified in this document.
- **Host** refers to the piece of general-purpose electronic computing equipment the device is connected or paired to, which can send data to and receive data from the device. It also hosts the MagneFlex Powder.

The word "user" is also often used in different ways in different contexts. In this document, user generally refers to the cardholder.

1.2 About the MagTek Powder

The MagneFlex Powder provides a convenient HTTP command interface to a device connected to a host. An HTTP client (such as a web browser) makes JSON calls to the host that are mapped to the device's low-level command set, as found in the Programmer's Reference. The MagneFlex Powder can be launched on the host as either a Windows Service, or a through a standalone executable. Any HTTP client that can reach the host via a network can process commands to the device. Commands are processed as simple request/response pairs, as shown later in the document.

The API also includes a sample SOAPUI project that demonstrates how JSON calls to the MagneFlex Powder are formed and processed. In addition, source code for the standalone executable is provided, if the developer wishes to integrate the MagneFlex Powder directly into their own code.

The MagneFlex Powder is single-threaded. If the service is busy processing a command to the device, other calls will be rejected.

1.3 SDK Contents

Executables:

File	Description
MTPPSCRA.WEBAPI.Host.exe	MagTek WEBAPI executable
MTPPSCRA.WEBAPI.HostService.exe	MagTek WEBAPI Windows service
MTPPSCRA.WEBAPI.Host.exe.config	MagTek WEBAPI executable configuration file
MTPPSCRA.WEBAPI.HostService.exe	MagTek WEBAPI Windows service configuration files

DLLs:

File	Description
MTDevice.DLL	MagTek PPSCRA Device constance library
MTLIB.DLL	MagTek PPSCRA interface library
MTPPSCRANET.DLL	MagTek PPSCRA library for .Net
MTPPSCRA.WEBAPI.DLL	MagTek PPSCRA library for WEB API
MTPPServiceNet.DLL	MagTek PPSCRA connection service library for .Net
MTEMVTLVParser.DLL	MagTek PPSCRA TLV EMV Parser
MTTLV.DLL	MagTek PPSCRA TLV library

Sample SOAPUI project:

File	Description
MTPPSCRA WEB API Sample-soapui-project.xml	Sample SOAPUI project file

1.4 System Requirements

Tested operating systems:
Windows 7
Windows 8
Windows 8.1
Windows 10

Microsoft .Net Framework 4.5 installed. (The API installation process will install this if it does not already exist on the host.)

Tested development environments:
Windows 8.1 with Microsoft Visual Studio 2013

1.5 Interfaces for Operating Systems

The following table matches the device interface to operating system.

Device	Interface	Operating System
DynaPro	USB	Windows 7, Windows 8, 8.1 & Windows 10
	ETHERNET	Windows 7, Windows 8, 8.1 & Windows 10
DynaPro Mini	USB	Windows 7, Windows 8, 8.1 & Windows 10
	BLE	Windows 8, 8.1 & Windows 10
DynaPro Go	USB	Windows 7, Windows 8, 8.1 & Windows 10
	BLE	Windows 8, 8.1 & Windows 10
	802.11 Wireless	Windows 8, 8.1 & Windows 10
IPAD	USB	Windows 7, Windows 8, 8.1 & Windows 10

2 How to Set Up the MagneFlex Powder

2.1 How to Connect MagneFlex Powder Service to a Host

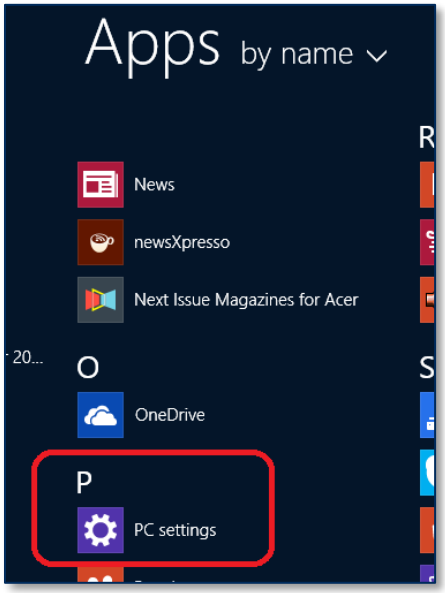
To use the MagneFlex Powder (MTPPSCRA.WEBAPI.HostService.exe)

1) In all Request, set the header ContentType to `"application/json"`
2) Build the JSON object for the MagneFlex Powder resource to be accessed.
3) Send HTTP request methods GET and POST (resource dependent) to the base address http://localhost:9000/api/mtppscrahost/ with the resource endpoint concatenated.

2.2 How to Connect DynaPro Mini to a Windows Host via BLE

To connect DynaPro Mini to a host with Windows 8.1 or higher and Bluetooth 4.0 hardware that supports BLE, follow these steps:

1) If you are using an external Bluetooth adapter, install any required drivers and connect it to the host.
2) On the host, install and configure the software you intend to use with DynaPro Mini:
   a) Make sure the host software is configured to look for the device on the proper connection.
   b) Make sure the host software knows which device(s) it should interface with.
   c) Make sure the host software is configured to properly interpret incoming data from the device. This depends on whether the device is configured to transmit data in GATT format or streaming format emulating a keyboard.
3) Make sure the DynaPro Mini has an adequate charge
4) Unpair from any other host it is already paired with before continuing.
5) Enter app mode, scroll down to **Apps by name**, and launch the Windows **PC Settings** app.



6) In the left side navigator, select **PC and devices** > **Bluetooth**.
7) Make sure Bluetooth is turned on and close the **PC and devices** app.
8) Launch the Windows **Manage Bluetooth Devices** app by following these steps:
   a) Enter desktop mode by swiping in from the left side of the touchscreen.

b) Touch the Bluetooth icon in the system tray and select **Add a Bluetooth Device** (see **Figure 2-1**).

Figure 2-1 - Launch Manage Bluetooth Devices App from Desktop Mode

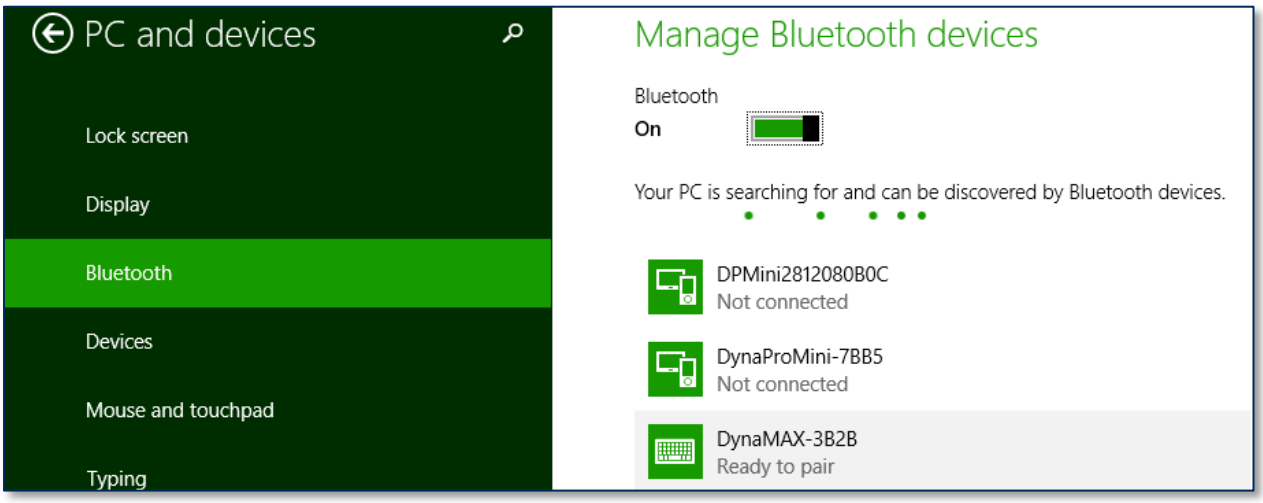
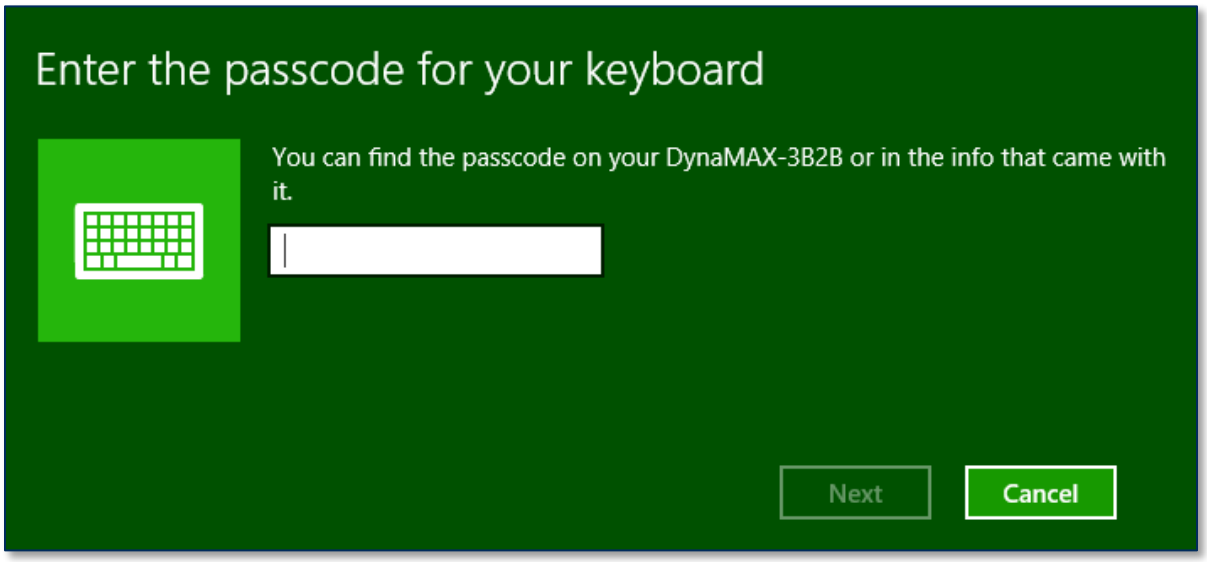
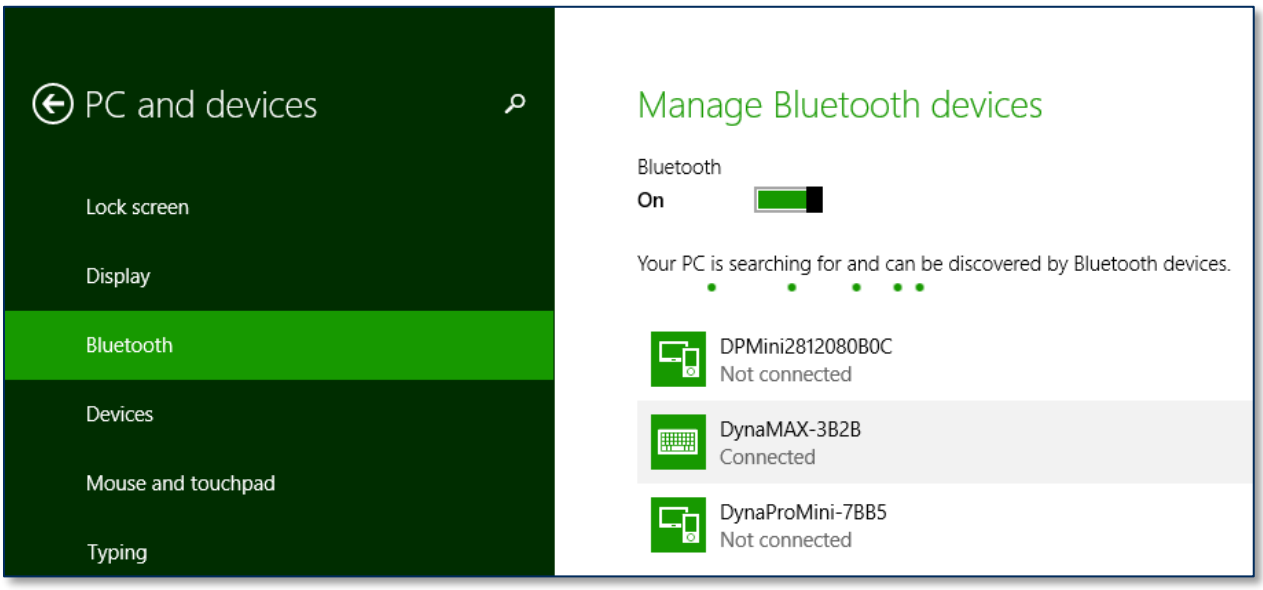


Figure 2-2 – Windows 8 Manage Bluetooth Devices App

9) Locate the serial number on the label on the bottom of the device. Note the final four digits.
10) Read through the list of pairable devices and locate the device called **DynaProMini-nnnn**, where nnnn is the last four digits of the device's serial number (if the device does not show in the list, power it off then power it back on). Below the device name you should see the text **Ready to pair**.
11) Select the device and press the **Pair** button. If the device is configured to run in KB mode, Windows will prompt you **Enter the passcode for your keyboard**.



12) Enter default passcode **000000** (or the device's actual password if it has been configured differently), then press the **Next** button. Windows will return you to the **Manage Bluetooth devices** page. After a short period of time, you will see the text **Connected** below the device you are pairing with. After a few seconds the device will disconnect, which is normal power-saving behavior.



13) Use the host software to test swiping a card. If you do not yet have host software and the device is configured to run in KB mode, open any text editor and swipe a card. The card contents should appear in the text editor.

14) The device consumes very little power when not transmitting card data, so it is not necessary to power off the device to conserve power. If the device appears as **Not connected** in the Windows list of Bluetooth devices, swiping a card should cause the device to reconnect briefly, transmit the card data, then disconnect.

15) Remember to change the default password. See the DynaPro Mini Programmer's Reference documents for details.

To unpair from the device:

1) Locate the device in the **Manage Bluetooth devices** window.

Press the **Remove device** button.

3 MagneFlex Powder Resources

MagneFlex Powder can be hosted as a Windows service (MagTek PPSCRA WEBAPI Host service or executable (MTPPSCRA.WEBAPI.Host.exe).

3.1 CheckHealth

Returns the operational status of the MagnFlex Powder.
Using Method GET:

```
api/mtppscrahost/CheckHealth
```

Return Value: The **CheckHealth Output**. A string array containing API name and status.

3.2 ReleaseDevice

Closes the connection to the device.
Using Method POST:

```
api/mtppscrahost/ReleaseDevice
```

Return Value:
None.

3.3 ReleaseDeviceEx

Closes the connection to the device and displays an idle bitmap message. This operation is not applicable on an device which was already closed.
Using Method POST:

```
api/mtppscrahost/ReleaseDeviceEx
```

Parameter (type)	Description
EndSessionDisplayMessage (int)	Display to show on the device. 0 = “Welcome” (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value:
None.

3.4 RequestCardSwipe

Prompt for a card swipe.
Using Method POST:

```
api/mtppscrahost/RequestCardSwipe
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.

WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
DisplayMessage (int)	Message to prompt the user with: 0 = Swipe Card / Idle alternating 1 = Swipe Card 2 = Please Swipe Card 3 = Please Swipe Card Again 4 = Chip Error, Use Mag Stripe
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps
FieldSeparator (string)	Delimiter to separate the output data.
CloseDevice (boolean)	Close the connection to the device after the request is processed. “false” = Do not close the device. (default) “true” = Close the device.
EndSession (boolean)	Send an EndSession command after the request is processed. “false” = Do not end the session. (default) “true” = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to “true”. 0 = “Welcome” (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **CardSwipe** Output.

```
{“CardSwipeOutput”: {}, “AdditionalOutputData”:{}}
```

3.5 RequestDeviceStatus

Retreives the device status.
Using Method POST:

```
api/mtppscrahost/RequestDeviceStatus
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
CreateNewConnection (boolean)	Create a new connection. “true” = Create new connection “false” = Do not create new connection

Return Value: The **DeviceStatus** Output output.

```
{"CardPresent": {}, "DeviceStatus":{}}
```

3.6 RequestEMVTags

Retrieves EMV tages from the device.
Using Method POST:

```
api/mtppscrahost/RequestEMVTags
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
TransactionType (int)	Transaction Type: 00 = Payment 01 = Cash 09 = Purchase with Cashback 20 = Refund
TagType (int)	EMV tag to set or get: 00 = Reader tags 80 = Application tags Lower 7 bits indicate which application slot of operation.
TagOperation (int)	Type of operation to be performed: 00 = Read single tag operation 03 = Read all tags operation 04 = Write operation FF = Set to factory defaults
DataBase (int)	Database Selector: 00 = Contact L2 EMV Tags 01 = PayPass-MasterCard 02 = PayWave-VISA 03 = ExpressPay-AMEX 04 = Discover

Data
(string)
	TLV data block to send to the device.
	Data block must be formed as an F9 CBC-MAC container message.
	Reference the device manual for details.
	`AAAA /* 2-byte MSB message length excluding padding and CBC-MAC */`
	`F9<len> /* container for MAC structure and generic data */`
	`DFDF55(MAC Encryption Type)<len><val>`
	`DFDF25(IFD Serial Number)<len><val>`
	`FA<len>/* container for generic data */`
	`<tag><len><val>`
	`...`
	`<tag><len><val>`
	`<Buffer if any to make blocks as multiple of 8 bytes>`
	`<CBC-MAC (4 bytes, use MAC variant of AMK)>`

RequestType (enum)	"SET" for setting an EMV tag. "GET" for getting an EMV tag.
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **CardSwipe** Output.

```
{"CardSwipeOutput": {}, "AdditionalOutputData":{}}
```

3.7 RequestManualSwipe

Promts the user to manually enter card data.
Using Method POST:

```
api/mtppscrahost/RequestManualSwipe
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)

Parameter (type)	Description
Options (int)	This is an ORed combination of flags that changes the device's data entry request behavior as follows: Bits 0 and 1 0 = Acct,Date,CVC 1 = Acct,Date 2 = Acct,CVC 3 = Acct Bit 2 1=Use QwickCodes entry Bit 3 1=Use PAN in PIN block creation Bit 4 0=Use PAN min 9, max 19 1=Use PAN min 14, max 21 Bits 5-7 are reserved and should be set to 0.
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **DeviceStatus** Output
Returned after RequestDeviceStatus.

Example :

```
{
   "CardPresent": false,
   "DeviceState": "00"
}
```

ManualSwipe Output.

```
{"CardManualOutput": {}, "StatusCode":, "AdditionalOutputData":{}}
```

3.8 RequestPIN

Prompts the user to enter a PIN by displaying one of five predetermined messages and plays a specified sound.
Using Method POST:

```
api/mtppscrahost/RequestPIN
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
PinMode (int)	Message to display as a user prompt: 0 = PINsgEnterPIN 1 = PINMsgEnterPINAmt 2 = PINMsgReenterPINAmt 3 = PINMsgReenterPIN 4 = PINMsgVerifyPIN
MaxPinLength (int)	Maximum PIN length. Must be less than 13.
MinPinLength (int)	Minimum PIN length. Must be greater than 3.
Options (int)	PIN verification and format: 0 = ISO0 Format, No verify PIN 1 = ISO3 Format, No verify PIN 2 = ISO0 Format, Verify PIN 3 = ISO3 Format, Verify PIN
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps
FieldSeparator (string)	Delimiter to separate the output data.
CloseDevice (boolean)	Close the connection to the device after the request is processed. “false” = Do not close the device. (default) “true” = Close not device.
EndSession (boolean)	Sends an EndSession command after the request is processed. “false” = Do not end the session. (default) “true” = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to “true”. 0 = “Welcome” (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **PIN** Output.

```
{"PINOutput": {"PINData":,}, "AdditionalOutputData":{}}
```

3.9 RequestSendCommand

Sends a command to the device and returns the raw response from the device.
Using Method POST:

```
api/mtppscrahost/RquestSendCommand
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
Data (string)	Hex string for command. Reference device manual for details.
RequestType (enum)	"SET" for commands where the ACK status is to be returned. "GET" for commands where the data is to be returned.
WaitForReport (string)	The report number to wait for before returning the response. Example: Command 30 00 (Set KSN) will respond with an ACK (01) or with data in Get Mode (30). WaitForReport = "01" will return the response for ACK report (01). WaitForReport = "30" will return the data report (30). Reference device manual for report numbers corresponding to a command request.
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display message to show on the device. 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **Data** Output in Hex string format of device raw response for this command.

```
{
   "Data":,
   "AdditionalOutputData": null
```

```
}
```

3.10 RequestSignature

Prompts the user to sign on the device's screen.
Using Method POST:

```
api/mtppscrahost/RequestSignature
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
Options (int)	Option to select the timeout behavior. 0 = Timeout will clear data 1 = Timeout with available data, signature can be retrieved if exists
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **Signature** Output.

```
{"SignatureOutput":{
      "SignatureOutputStatus":,
      "SignatureData":} ,
 "AdditionalOutputData":{}}
```

3.11 RequestSmartCard

Begins an EMV transaction.
Using Method POST:

```
api/mtppscrahost/RequestSmartCard
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
CardType (int)	Card type that can be used for the transaction: 1 = Magnetic stripe 2 = Contact smart card 3 = Magnetic stripe or contact smart card 4 = Contactless smart card (Not supported on DynaPro Mini) 5 = Contactless smart card + magnetic stripe 6 = Contactless smart card + contact smart card 7 = Magnetic stripe + contact smart card + contactless smart card.
ConfirmationWaitTime (int)	Time the device will wait for the user to begin the transaction.
PINEntryWaitTime (int)	Time the device will wait for the user to enter the PIN.
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps
Options (int)	Transaction options: 0 = Normal 1 = Bypass PIN 2 = Force Online 4 = Acquirer not available
TransactionType (int)	Type of transaction to be used: 0x02 = Cash back 0x04 = Goods 0x08 = Services
Amount (decimal)	The amount to be used and authorized in decimal format. 1.01 = 1 dollar and 1 cent
CashBack (decimal)	The amount of cashback to be used and authorized in decimal format. 1.01 = 1 dollar and 1 cent
QwickChipMode (boolean)	Start the transaction in Qwick Chip mode. In Qwick Chip mode, the devices does not prompt for an amount approval and needs no Acquirer Response (RequestSendAcquirerResponse). “true” = Qwick chip mode. “false” = Do not do qwick chip mode.
Reserved (int)	Reserved for future use. These are the reserved bytes begining at index 20 of the 0xA2 command. Reference device manual ***D99875585*** for more details. Example: 00000000000000000000000084000**01**00 – QuickDip Mode 00000000000000000000000084000000**03** – PIN Verify 00000000000000000000000084000000**04** – PIN Set

Parameter (type)	Description
AdditionalRequestData (array)	Additional key/value pairs of data
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **Transaction** Output SmartCard for SmartCard.

```
{"TransactionOutput": {}}
```

3.12 RequestSmartCardEx

Begins an EMV transaction. The response contains two separate transaction outputs. The first is ARQC (DataType1) and the second is Batch data (DataType2).
Using Method POST:

```
api/mtppscrahost/RequestSmartCard
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
CardType (int)	Card type that can be used for the transaction: 1 = Magnetic stripe 2 = Contact smart card 3 = Magnetic stripe or contact smart card 4 = Contactless smart card (Not supported on DynaPro Mini) 5 = Contactless smart card + magnetic stripe 6 = Contactless smart card + contact smart card 7 = Magnetic stripe + contact smart card + contactless smart card.
ConfirmationWaitTime (int)	Time the device will wait for the user to begin the transaction.
PINEntryWaitTime (int)	Time the device will wait for the user to enter the PIN.
Tones (int)	Tones to use: 0 = No sound 1 = One beep 2 = Two beeps

Parameter (type)	Description
Options (int)	Transaction options: 0 = Normal 1 = Bypass PIN 2 = Force Online 4 = Acquirer not available
TransactionType (int)	Type of transaction to be used: 0x02 = Cash back 0x04 = Goods 0x08 = Services
Amount (decimal)	The amount to be used and authorized in decimal format. 1.01 = 1 dollar and 1 cent
CashBack (decimal)	The amount of cashback to be used and authorized in decimal format. 1.01 = 1 dollar and 1 cent
QwickChipMode (boolean)	Start the transaction in Qwick Chip mode. In Qwick Chip mode, the devices does not prompt for an amount approval and needs no Acquirer Response (RequestSendAcquirerResponse). "true" = Qwick chip mode. "false" = Do not do qwick chip mode.
Reserved (int)	Reserved for future use. These are the reserved bytes begining at index 20 of the 0xA2 command. Reference device manual ***D99875585*** for more details. Example: 00000000000000000000000084000**01**00 – QuickDip Mode 00000000000000000000000084000000**03** – PIN Verify 00000000000000000000000084000000**04** – PIN Set
AdditionalRequestData (array)	Additional key/value pairs of data
CloseDevice (boolean)	Close the connection to the device after the request is processed. "false" = Do not close the device. (default) "true" = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. "false" = Do not end the session. (default) "true" = End the session.
EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **Transaction** Output SmartCard for SmartCard.

```
{"TransactionOutput": {}}
```

3.13 RequestSendAcquirerResponse

Sends the ARPC to the device. Applicable only after a RequestSmartCard with QwickChipMode set to false.
Using Method POST:

```
api/mtppscrahost/RequestSendAcquirerResponse
```

Parameter (type)	Description
DeviceID (string)	URI of the device. See **DeviceID URI** for details.
WaitTime (int)	Time in seconds the device will wait for the action to be completed. (1 - 255)
WaitTimeBeforeTransactionComplete (int)	Time in seconds to wait after receiving the transaction response before the transaction is complete.
IssuerAuthenticationData (string)	Issuer response to the transaction request in hexadecimal format. This field is for the data portion of the EVM Tag 91.
IssuerScriptTemplate1 (string)	Issuer Script to send to ICC in hexadecimal format. This field is for the data portion of the EVM Tag 71.
IssuerScriptTemplate2 (string)	Issuer Script to send to ICC in hexadecimal format. This field is for the data portion of the EVM Tag 72.
ApprovalStatus (int)	Status from acquirer/issuer. This field represents the data portion of the EMV Tag 8A. Example: 0 – Approve 1 – Decline
KSN (string)	Key serial number used for the transaction
DeviceSerialNumber (string)	Device serial number
AdditionalRequestData (array)	Additional key/value pairs of data
CloseDevice (boolean)	Close the connection to the device after the request is processed. “false” = Do not close the device. (default) “true” = Close the device.
EndSession (boolean)	Sends an EndSession command after the request is processed. “false” = Do not end the session. (default) “true” = End the session.

EndSessionDisplayMessage (int)	Display to show on the device when EndSession is set to "true". 0 = "Welcome" (default) 1 = Bitmap Slot 1 2 = Bitmap Slot 2 3 = Bitmap Slot 3 4 = Bitmap Slot 4

Return Value: The **Transaction** Output SmartCardEx
Returned after a RequestSmartCardEx.

Example:


```
{"TransactionOutput": [
      {
      "TransactionType": 0,
      "TransactionStatus": 255,
      "OperationStatus": 0,
      "DataType": 1,
      "Data":
"012EF982012ADFDF540A9010010B999999000080DFDF550182DFDF2508992B8088191
60710FA82010770820103DFDF5301005F2014434F4E544143544C4553532F4D4147544
54B20445F30020201DFDF4D273B353332353030303035303030333435353D323530393
030303030303030303030303030303FDFDF520105F88200AEDFDF59820090AFA55EB
7924AE0370697A1DBAFA44AA75A9D4495597DF5C9AE2A323899497B1A572BAE9094F65
1274495CCDAD078D497E10045528BADF025AFEF677C63D8D332190E870FDE461E9D5CC
F0EA882B6F17D655A6368FFE20BD1400899810CC70D5442C06443650134322AA900CFB
DEED7BCF95CB5E79CCCE90D0D33EA13EE4EC7436B041C2FC3837C1F55C9151763F148D
2DFDF560A9010010B999999000080DFDF570180DFDF580102D038DCA5",
      "RawData":
"AS75ggEq399UCpAQAQuZmZkAAIDf31UBgt/fJQiZK4CIGRYHEPqCAQdwggED399TAQBfI
BRDT05UQUNUTEVTUy9NQUdURUsgRF8wAgIB399NJzs1MzI1MDAwMDUwMDAzNDU1PTI1MDk
wMDAwMDAwMDAwMDAwMDAwP9/fUgEF+IIArt/fWYIAkK+lXreSSuA3Bpeh26+kSqdanUSVW
X31ya4qMjiZSXsaVyuukJT2USdElcza0HjUl+EARVKLrfAlr+9nfGPY0zIZDocP3kYenVz
PDqiCtvF9ZVpjaP/iC9FACJmBDMcNVELAZENlATQyKqkAz73u17z5XLXnnMzpDQ0z6hPuT
sdDawQcL8ODfB9VyRUXY/FI0t/fVgqQEAELmZmZAACA399XAYDf31gBAtA43KU=",
      "KSN": "9010010B999999000080",
      "DeviceSerialNumber": "992B808819160710",
      "EncryptionType": "80",
      "NumberOfPaddedBytes": 2,
      "EMVSREDData":
"AFA55EB7924AE0370697A1DBAFA44AA75A9D4495597DF5C9AE2A323899497B1A572BA
E9094F651274495CCDAD078D497E10045528BADF025AFEF677C63D8D332190E870FDE4
61E9D5CCF0EA882B6F17D655A6368FFE20BD1400899810CC70D5442C06443650134322
AA900CFBDEED7BCF95CB5E79CCCE90D0D33EA13EE4EC7436B041C2FC3837C1F55C9151
763F148D2",
      "MerchantData": "",
      "FallbackIndicator": "00",
      "MaskedICCTrack2":
"3B353332353030303035303030333435353D32353039303030303030303030303030303
03030303F",
      "ServiceCode": "0201",
```

```
      "CardHolderName": "434F4E544143544C4553532F4D414754454B2044",
      "CardType": "05",
      "ApplicationIdentifier": null,
      "NumberOfPaddedBytesForBatch": 0,
      "EMVSREDDataForBatch": null
   },
      {
      "TransactionType": 1,
      "TransactionStatus": 1,
      "OperationStatus": 0,
      "DataType": 2,
      "Data":
"01A3F982019FDFDF540A9010010B999999000080DFDF550182DFDF2508992B8088191
60710FA82017CF0820178F105DFDF1A0101F88200DEDFDF598200C05F4D99FEED24282
571B3AC926F3F9BA142B39805452A3F7C8F49198B298A2FC41E2C382C5CCD894FD5ECE
4E232A8D70E57E39FFBB6AC30E94250C9BC2559BB6ACBBF9E7C185B1FD16F9CE2D79EB
73CA58439E67BDDCDA8AF257C9E42165CE069B129E53A7AB5BAC22CF4392E4E274EC42
52CF539C8B43B473BCDCD1D21419D7E353CD7E7CFFA624BACC2A23FA633E18D63C2586
EC555C698B0AC3888764A6F7679FE1D6CE975A7EE2AF5A14040B2A6014E5D18D5B109F
7B672F2A7ADE1C5B90ADFDF560A9010010B999999000080DFDF570180DFDF580100F78
2008BDFDF4001005F25032009015F24032509305F2A0208409F02060000000001009F0
607A00000000410109F1C0831313232333334349F3901059C01009F34034203005F570
1305F3401015F2014434F4E544143544C4553532F4D414754454B2044DFDF4D273B353
33235303030303530303033435353D32353039303030303030303030303030303030303
03F000000D5950C1A",
      "RawData":
"AaP5ggGf399UCpAQAQuZmZkAAIDf31UBgt/fJQiZK4CIGRYHEPqCAXzwggF48QXf3xoBA
fiCAN7f31mCAMBfTZn+7SQoJXGzrJJvP5uhQrOYBUUqP3yPSRmLKYovxB4sOCxczYlP1ez
k4jKo1w5X45/7tqww6UJQybwlWbtqy7+efBhbH9FvnOLXnrc8pYQ55nvdzaivJXyeQhZc4
GmxKeU6erW6wiz0OS5OJ07EJSz1Oci0O0c7zc0dIUGdfjU81+fP+mJLrMKiP6Yz4Y1jwlh
uxVXGmLCsOIh2Sm92ef4dbOl1p+4q9aFAQLKmAU5dGNWxCfe2cvKnreHFuQrf31YKkBABC
5mZmQAAgN/fVwGA399YAQD3ggCL399AAQBfJQMgCQFfJAMlCTBfKgIIQJ8CBgAAAAABAJ8
GB6AAAAAEEBCfHAgxMTIyMzM0NJ85AQWcAQCfNANCAwBfVwEwXzQBAV8gFENPTlRBQ1RMR
VNTL01BR1RFSyBE399NJzs1MzI1MDAwMDUwMDAzNDU1PTI1MDkwMDAwMDAwMDAwMDAwMDA
wPwAAANWVDBo=",
      "KSN": "9010010B999999000080",
      "DeviceSerialNumber": "992B808819160710",
      "EncryptionType": "80",
      "NumberOfPaddedBytes": 0,
      "EMVSREDData":
"5F4D99FEED24282571B3AC926F3F9BA142B39805452A3F7C8F49198B298A2FC41E2C3
82C5CCD894FD5ECE4E232A8D70E57E39FFBB6AC30E94250C9BC2559BB6ACBBF9E7C185
B1FD16F9CE2D79EB73CA58439E67BDDCDA8AF257C9E42165CE069B129E53A7AB5BAC22
CF4392E4E274EC4252CF539C8B43B473BCDCD1D21419D7E353CD7E7CFFA624BACC2A23
FA633E18D63C2586EC555C698B0AC3888764A6F7679FE1D6CE975A7EE2AF5A14040B2A
6014E5D18D5B109F7B672F2A7ADE1C5B90A",
      "MerchantData":
"DFDF4001005F25032009015F24032509305F2A0208409F02060000000001009F0607A
00000000410109F1C0831313232333334349F3901059C01009F34034203005F5701305
F3401015F2014434F4E544143544C4553532F4D414754454B2044DFDF4D273B3533323
```

```
5303030303530303033435353D3235303930303030303030303030303030303F"
,
      "FallbackIndicator": "",
      "MaskedICCTrack2":
"3B353332353030303035303030333435353D3235303930303030303030303030303
03030303F",
      "ServiceCode": "",
      "CardHolderName": "434F4E544143544C4553532F4D414754454B2044",
      "CardType": "",
      "ApplicationIdentifier": "A0000000041010",
      "NumberOfPaddedBytesForBatch": 0,
      "EMVSREDDataForBatch": null
   }
]}
```


Transaction Output ARPC for ARPC.

```
{"TransactionOutput": {}}
```

3.14 RequestOperationStatus

Retrieves the operation status of the device.
Using Method POST:

```
api/mtppscrahost/RquestOperationStatus
```

Return Value: The **OperationStatus** Output output.

```
{
   "OperationStatus": ,
   "DeviceID": null,
   "AdditionalOutputData": null
}
```

4 MagneFlex Powder Response Output Structures

The MagneFlex Powder returns the following outputs.

4.1 CheckHealth Output

Returned after CheckHealth. A string array containing API name and status.

Example:

```
[
   "MagTek PPSCRA WEB API",
   "OK"
]
```

4.2 CardSwipe Output

Returned after RequestCardSwipe.

Example :


```
{
   "CardSwipeOutput":    {
      "CardOperationStatus": 0,
      "CardStatus": 0,
      "CardType": 1,
      "DataType": 34,
      "EncryptedMagnePrint":
"787E070A18ECF5087595FD4CDE53E550051E719BDCBCF29C7646D6B5AF90EEEEA8871
3ADE97F118095C8CF3C36426FAD860E1BF5C3465D21",
      "EncryptedTrack1":
"C5389DFDB735F9D9EF628E0FDC446FB9CCBEBE959DECD22655737B41CD249D6C67244
6BF77A3738D09FBF8DD55A61906CB50C697994344DDFE657C3F8ED287BD4141788C400
BAEFBA0E52BE1498AE186",
      "EncryptedTrack2":
"12BA7B8F7386EB4ED1EAD4467E621E26930A269FA733607B73AEE894F8F9B5A336706
6B822881FE5",
      "EncryptedTrack3":
"12BA7B8F7386EB4ED1EAD4467E621E26F727B4225A3C8F73351BEF09B1ED74971CE9A
1B4F84A1440109735A65864FAA2384D29912454DE6B37485C4060D8809AD7A00C24F2D
78A340CEBDD00CE0AF419BAC7C9F89DD4415D45B578B962EE4126D6EB943ABC5D729B4
5E7C09060B3E3AB",
      "EncryptedMagnePrintLength": 56,
      "EncryptedMagnePrintStatus": 0,
      "EncryptedTrack1Length": 80,
      "EncryptedTrack1Status": 0,
      "EncryptedTrack2Length": 40,
      "EncryptedTrack2Status": 0,
      "EncryptedTrack3Length": 112,
      "EncryptedTrack3Status": 0,
      "MagStripeStatus": 0,
      "PANDataLength": 32,
      "Track1Length": 78,
      "Track1Status": 0,
```

```
      "Track2Length": 39,
      "Track2Status": 0,
      "Track3Length": 106,
      "Track3Status": 0,
      "StatusCode": 0,
      "CardData":
"CardType=1|OperationStatus=0|CardStatus=0|DataType=34|Track1Status=0|
Track1Length=78|Track1=%B4019000000000003000^FIREMAN/TEST D
^161000000000000000000000000?|Track2Status=0|Track2Length=39|Track2=;
4019000000000003000=16100000000000000?|Track3Status=0|Track3Length=106
|Track3=;4019000000000003000=00000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000?|EncTrack1Status=0|EncTrac
k1Length=80|EncTrack1=C5389DFDB735F9D9EF628E0FDC446FB9CCBEBE959DECD226
55737B41CD249D6C672446BF77A3738D09FBF8DD55A61906CB50C697994344DDFE657C
3F8ED287BD4141788C400BAEFBA0E52BE1498AE186|EncTrack2Status=0|EncTrack2
Length=40|EncTrack2=12BA7B8F7386EB4ED1EAD4467E621E26930A269FA733607B73
AEE894F8F9B5A3367066B822881FE5|EncTrack3Status=0|EncTrack3Length=112|E
ncTrack3=12BA7B8F7386EB4ED1EAD4467E621E26F727B4225A3C8F73351BEF09B1ED7
4971CE9A1B4F84A1440109735A65864FAA2384D29912454DE6B37485C4060D8809AD7A
00C24F2D78A340CEBDD00CE0AF419BAC7C9F89DD4415D45B578B962EE4126D6EB943AB
C5D729B45E7C09060B3E3AB|EncMPStatus=0|EncMPLength=56|EncMP=787E070A18E
CF5087595FD4CDE53E550051E719BDCBCF29C7646D6B5AF90EEEEA88713ADE97F11809
5C8CF3C36426FAD860E1BF5C3465D21|MPSTS=00002200|MSStatus=0|KSN=95000300
00000120073D|SerialNumber=98D90C660E070F0E|PAN=12BA7B8F7386EB4ED1EAD44
67E621E26930A269FA733607B32040311C324348B|CBCMAC=7202CDF1",
      "CBCMAC": "7202CDF1",
      "KSN": "95000300000000120073D",
      "MagnePrintStatus": "00002200",
      "PANData":
"12BA7B8F7386EB4ED1EAD4467E621E26930A269FA733607B32040311C324348B",
      "MagTekSerialNumber": "98D90C660E070F0E",
      "Track1": "%B4019000000000003000^FIREMAN/TEST D
^161000000000000000000000000?",
      "Track2": ";4019000000000003000=16100000000000000?",
      "Track3":
";4019000000000003000=00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000?"
   },
   "AdditionalOutputData": null
}
```


4.3 DeviceStatus Output

Returned after RequestDeviceStatus.

Example :


```
{
   "CardPresent": false,
   "DeviceState": "00"
}
```

4.4 ManualSwipe Output

Returned after RequestManualSwipe.

Example :


```
{
   "CardManualOutput":    {
      "CardOperationStatus": 0,
      "CardStatus": 0,
      "CardType": 3,
      "DataType": 34,
      "EncryptedMagnePrint": "",
      "EncryptedTrack1":
"D5FCF8CAA7EB399A13D097C58D6D39123EA41CDCB45E3C3C4FAF1741984114DC5BD60
6E2AD201321C00E08A359B42D7CB4EA09B8853D11161E505BB7EF621CDD",
      "EncryptedTrack2":
"819EAA4AF1962A3BD08AEC151002BB03C42D731AAD37CC76DDD8BCDE0F93963BA9CED
1B157D630D4",
      "EncryptedTrack3": "",
      "EncryptedMagnePrintLength": 0,
      "EncryptedMagnePrintStatus": 1,
      "EncryptedTrack1Length": 64,
      "EncryptedTrack1Status": 0,
      "EncryptedTrack2Length": 40,
      "EncryptedTrack2Status": 0,
      "EncryptedTrack3Length": 0,
      "EncryptedTrack3Status": 1,
      "MagStripeStatus": 0,
      "PANDataLength": 32,
      "Track1Length": 57,
      "Track1Status": 0,
      "Track2Length": 39,
      "Track2Status": 0,
      "Track3Length": 0,
      "Track3Status": 1,
      "StatusCode": 0,
      "CardData":
"CardType=3|OperationStatus=0|CardStatus=0|DataType=34|Track1Status=0|
Track1Length=57|Track1=%M1111000004000001111^MANUAL
ENTRY/^22220000000000000000?|Track2Status=0|Track2Length=39|Track2=;11
11000004000001111=22220000000000000?|Track3Status=1|Track3Length=0|Tra
ck3=|EncTrack1Status=0|EncTrack1Length=64|EncTrack1=D5FCF8CAA7EB399A13
D097C58D6D39123EA41CDCB45E3C3C4FAF1741984114DC5BD606E2AD201321C00E08A3
59B42D7CB4EA09B8853D11161E505BB7EF621CDD|EncTrack2Status=0|EncTrack2Le
ngth=40|EncTrack2=819EAA4AF1962A3BD08AEC151002BB03C42D731AAD37CC76DDD8
BCDE0F93963BA9CED1B157D630D4|EncTrack3Status=1|EncTrack3Length=0|EncTr
ack3=|EncMPStatus=1|EncMPLength=0|EncMP=|MPSTS=00000000|MSStatus=0|KSN
=95000300000000120073E|SerialNumber=98D90C660E070F0E|PAN=819EAA4AF1962A
3BD08AEC151002BB03C42D731AAD37CC7673CBBC8B99470A82|CBCMAC=35F8A181",
      "CBCMAC": "35F8A181",
      "KSN": "95000300000000120073E",
      "MagnePrintStatus": "00000000",
```

```
      "PANData":
"819EAA4AF1962A3BD08AEC151002BB03C42D731AAD37CC7673CBBC8B99470A82",
      "MagTekSerialNumber": "98D90C660E070F0E",
      "Track1": "%M1111000004000001111^MANUAL
ENTRY/^22220000000000000000?",
      "Track2": ";1111000004000001111=22220000000000000?",
      "Track3": ""
   },
   "StatusCode": 0,
   "AdditionalOutputData": null
}
```


4.5 PIN Output

Returned after RequestPIN.

The PINData key contains a comma delimited value:

Field Name	Value
PIN KSN	PIN Key serial number
EPB	Encrypted PIN block
opStatus	Operation status

Example:


```
{
   "PINOutput": {"PINData":"9A006300000001200065,59D0274E9F58DD5F,0"},
   "AdditionalOutputData": null
}
```


4.6 Signature Output

Returned after RequestSignature.

Example:


```
{
   "SignatureOutput":    {
      "SignatureOperationStatus": 0,
      "SignatureData":
"FF0UXBVcFVsVWhVZFlgWVxZWFlUWVBdTF1EYThhNGEwYShhJGEcZRRlEGUMZQRlAGT8ZP
hk9GjwaOxo6GjkbORs6GzscPBw9HD4dPx1BHkMfRh9HIEggSSBKIU0iTiJPI1AjUiRTJFQ
lVCVVJlUnVShVKFQoUylSKVEpUCpPKk4qTCtKK0gsRixELEEtPy09LTwtOi44LjcuNi41L
jQuMy40LjUuNi43LzgvOS86LzwvPS9AMEEwQzBFMEgxSjFLMU0yTzJRM1Q0VjRXNFg1WDV
ZNlk3WDhXOVU5VDpTOlI7UTxOPU09TD5JQEdBRUFEQ0JEQURART9GPkc9SDxIO0k6SjhLO
Es3TDdMNk01TjROM04yTjNNM0w0SzVKNko3SThIOUg6RztGPEU/REBDQUJDQkRARz9JPks
8TjxPO1E6UTlSOVP//0Y1RjZGN0c3RzhHOUc6SDtIPEg9ST5JQElBSkNLRUtGTEhMSk5NT
k9QUlBTUldTWFNZVFtVXVZeV19YX1hg//9PTk5OTk1NTUxMS0xKTEdKRkpESUNJQUhASD5
HPUc7RzlGNkY1Rv//bDFrMGowaTBoMGcwZjBkMGMwYjFhMWAyXjNdNFw1WzZaN1o4WTlZO
lg8Vz1XPlY/VkBVQlVEVUZUR1RJVEpUTFROVE9VUVVTVlRXVVdWWFdZV1pYW1lcWV5ZX1l
hWWJZZFhlWGZXZ1doVmpVa1VsVG1TblJvUnBRcVFyUHJPdE90TnVOdk12THdMd0t3SndJd
0h3R3dGd0V3RHdDd0J3QXdAdkB1QHRAc0ByQHFAcEBvQG5BbUFsQWtBa0JqQmlCaUNoQ2d
```

```
DZkRlRGREZEVjRWJFYkZhRmBGX0dfRv//fC18LnwvfS5+Ln8ugC2CLYMthC2FLIYsiCyJL
IoriyuMK44rjyuRK5MrlCuULJUsliz//4otiy2LLIsriiqKKYoqiSqJK4ksiS2JLokviTC
IM4g2iDeIO4g+iECIQYhDiEaJSIlLiUyJTYpOik+LUP//nCydLZ0unS+dMJ0xnTKdNJ01n
TadN505nTuePJ49nkCeQZ5EnkWeRp5HnkieSZ9K//+fLqAuoS6iLaMtpC2lLaUspiyoK6k
rqiurKqwqrSquKa8q//+gP6A+oD2hPaE8ojyjO6Q7pTunOqg6qjqrOaw5rTmuOa86sDqxO
///n0z//59MoEygS6FLoUqiSqNKpEmmSKdIqEepR6tGrEatRq9FsUX//8EswSvBKsErwCv
ALMAtwC7AL78wvzK/M780vza/Ob86vzy/Pb8/v0K/Rb9Hv0i/Sb9KwEv//788wDzBPMI7w
zrEOsQ5xTnGOMg3yTbKNcs0zDTNM88y0THSMNMv1C/VLtYu///DPsM9xD3EPsU+xj/HP8h
AyUHLQsxDzkTPRdBF0kfTSNVK1krXTNhN2U7aUP//"
   },
   "AdditionalOutputData": null
}
```


4.7 Transaction Output SmartCard

Returned after RequestSmartCard.

Example:


```
{"TransactionOutput": {
   "TransactionType": 0,
   "TransactionStatus": 255,
   "OperationStatus": 0,
   "DataType": 1,
   "ARQCData":
"01A4F98201A0DFDF540A9500030000000120071BDFDF550182DFDF250898D90C660E0
70F0EFA82017D70820179DFDF5301005F201A5649534120414351554952455220544553
3542F434152442030325F30020201DFDF4D273B343432373030303030393030303033
33333373D3232313230303030303030303030303030303FDFDF520105F882011EDFDF598
20100B00A0A84B0C31AF16B79E3309DC808B34DC05F71F094C78FC8A81F074EA3E5033
2ED61FC3921D319AA76E45CD26466A4F46E3B3C7D1D0280A0C395D66DF4010D1D0A59A
E65A7276AD0E764BFB34A44A514B32664710A8176F8771737B6B3D0F82F65B6F2375F7
5C575513F658F9749A01903F37E3E7DD09F1B048A8077A9017677879F4146DD15F211F
2967624CD8C41C1662F0F9BF4D842370125AA44114E286F8CD51D31C8E6F4774A74096
EECB06437DC07EFA01F5846DF751D13F5547A0E016A9AB51B84146365D9B501D608F7C
44A5880AC5FBBBC2ECF6EFD1BA141644FC27578C677AF5FBF21D6E3C65C77233CA7F7C
7A250C2E289597F9FFD224854F0DFDF560A9500030000000120071BDFDF570180DFDF5
801060000F3E97338",
   "BatchData": null,
   "RawARQCData":
"AaT5ggGg399UCpUAAwAAAAEgBxvf31UBgt/fJQiY2QxmDgcPDvqCAX1wggF5399TAQBfI
BpWSVNBIEFDUVVJUkVSIFRFU1QvQ0FSRCAwMl8wAgIB399NJzs0NDI3MDAwMDA5MDAwMDA
zMzM3PTIyMTIwMDAwMDAwMDAwMDAwP9/fUgEF+IIBHt/fWYIBALAKCoSwwxrxa3njMJ3IC
LNNwF9x8JTHj8ioHwdOo+UDMu1h/Dkh0xmqduRc0mRmpPRuOzx9HQKAoMOV1m30AQ0dClm
uZacnatDnZL+zSkSlFLMmZHEKgXb4dxc3trPQ+C9ltvI3X3XFdVE/ZY+XSaAZA/N+Pn3Qn
xsEioB3qQF2d4efQUbdFfIR8pZ2JM2MQcFmLw+b9NhCNwElqkQRTihvjNUdMcjm9HdKdAl
u7LBkN9wH76AfWEbfdR0T9VR6DgFqmrUbhBRjZdm1AdYI98RKWICsX7u8Ls9u/RuhQWRPw
nV4xnevX78h1uPGXHcjPKf3x6JQwuKJWX+f/SJIVPDf31YKlQADAAAAASAHG9/fVwGA399
YAQYAAPPpczg=",
   "RawBatchData": null,
   "KSN": "95000300000000120071B",
   "DeviceSerialNumber": "98D90C660E070F0E",
   "EncryptionType": "80",
```

```
   "NumberOfPaddedBytes": 6,
   "EMVSREDData":
"B00A0A84B0C31AF16B79E3309DC808B34DC05F71F094C78FC8A81F074EA3E50332ED6
1FC3921D319AA76E45CD26466A4F46E3B3C7D1D0280A0C395D66DF4010D1D0A59AE65A
7276AD0E764BFB34A44A514B32664710A8176F8771737B6B3D0F82F65B6F2375F75C57
5513F658F9749A01903F37E3E7DD09F1B048A8077A9017677879F4146DD15F211F2967
624CD8C41C1662F0F9BF4D842370125AA44114E286F8CD51D31C8E6F4774A74096EECB
06437DC07EFA01F5846DF751D13F5547A0E016A9AB51B84146365D9B501D608F7C44A5
880AC5FBBBC2ECF6EFD1BA141644FC27578C677AF5FBF21D6E3C65C77233CA7F7C7A25
0C2E289597F9FFD224854F0",
   "MerchantData": null,
   "FallbackIndicator": "00",
   "MaskedICCTrack2":
"3B343432373030303030393030303030333333373D3232313230303030303030303
03030303F",
   "ServiceCode": "0201",
   "CardHolderName":
"5649534120414351554952455220544553542F43415244203032",
   "CardType": "05",
   "ApprovalStatus": -1
}}
```


4.8 Transaction Output SmartCardEx

Returned after a RequestSmartCardEx.

Example:


```
{"TransactionOutput": [
      {
      "TransactionType": 0,
      "TransactionStatus": 255,
      "OperationStatus": 0,
      "DataType": 1,
      "Data":
"012EF982012ADFDF540A9010010B999999000080DFDF550182DFDF2508992B8088191
60710FA82010770820103DFDF5301005F2014434F4E544143544C4553532F4D4147544
54B20445F30020201DFDF4D273B353332353030303035303030333435353D323530393
03030303030303030303030303030303FDFDF520105F88200AEDFDF59820090AFA55EB
7924AE0370697A1DBAFA44AA75A9D4495597DF5C9AE2A323899497B1A572BAE9094F65
1274495CCDAD078D497E10045528BADF025AFEF677C63D8D332190E870FDE461E9D5CC
F0EA882B6F17D655A6368FFE20BD1400899810CC70D5442C06443650134322AA900CFB
DEED7BCF95CB5E79CCCE90D0D33EA13EE4EC7436B041C2FC3837C1F55C9151763F148D
2DFDF560A9010010B999999000080DFDF570180DFDF580102D038DCA5",
      "RawData":
"AS75ggEq399UCpAQAQuZmZkAAIDf31UBgt/fJQiZK4CIGRYHEPqCAQdwggED399TAQBfI
BRDT05UQUNUTEVTUy9NQUdURUsgRF8wAgIB399NJzs1MzI1MDAwMDUwMDAzNDU1PTI1MDk
wMDAwMDAwMDAwMDAwMDAwP9/fUgEF+IIArt/fWYIAkK+lXreSSuA3Bpeh26+kSqdanUSVW
X31ya4qMjiZSXsaVyuukJT2USdElcza0HjUl+EARVKLrfAlr+9nfGPY0zIZDocP3kYenVz
PDqiCtvF9ZVpjaP/iC9FACJmBDMcNVELAZENlATQyKqkAz73u17z5XLXnnMzpDQ0z6hPuT
sdDawQcL8ODfB9VyRUXY/FI0t/fVgqQEAELmZmZAACA399XAYDf31gBAtA43KU=",
      "KSN": "9010010B999999000080",
      "DeviceSerialNumber": "992B808819160710",
```

```
      "EncryptionType": "80",
      "NumberOfPaddedBytes": 2,
      "EMVSREDData":
"AFA55EB7924AE0370697A1DBAFA44AA75A9D4495597DF5C9AE2A323899497B1A572BA
E9094F651274495CCDAD078D497E10045528BADF025AFEF677C63D8D332190E870FDE4
61E9D5CCF0EA882B6F17D655A6368FFE20BD1400899810CC70D5442C06443650134322
AA900CFBDEED7BCF95CB5E79CCCE90D0D33EA13EE4EC7436B041C2FC3837C1F55C9151
763F148D2",
      "MerchantData": "",
      "FallbackIndicator": "00",
      "MaskedICCTrack2":
"3B353332353030303035303030333435353D323530393030303030303030303030303
03030303F",
      "ServiceCode": "0201",
      "CardHolderName": "434F4E544143544C4553532F4D414754454B2044",
      "CardType": "05",
      "ApplicationIdentifier": null,
      "NumberOfPaddedBytesForBatch": 0,
      "EMVSREDDataForBatch": null
   },
      {
      "TransactionType": 1,
      "TransactionStatus": 1,
      "OperationStatus": 0,
      "DataType": 2,
      "Data":
"01A3F982019FDFDF540A9010010B999999000080DFDF550182DFDF2508992B8088191
60710FA82017CF0820178F105DFDF1A0101F88200DEDFDF598200C05F4D99FEED24282
571B3AC926F3F9BA142B39805452A3F7C8F49198B298A2FC41E2C382C5CCD894FD5ECE
4E232A8D70E57E39FFBB6AC30E94250C9BC2559BB6ACBBF9E7C185B1FD16F9CE2D79EB
73CA58439E67BDDCDA8AF257C9E42165CE069B129E53A7AB5BAC22CF4392E4E274EC42
52CF539C8B43B473BCDCD1D21419D7E353CD7E7CFFA624BACC2A23FA633E18D63C2586
EC555C698B0AC3888764A6F7679FE1D6CE975A7EE2AF5A14040B2A6014E5D18D5B109F
7B672F2A7ADE1C5B90ADFDF560A9010010B999999000080DFDF570180DFDF580100F78
2008BDFDF4001005F25032009015F24032509305F2A0208409F02060000000001009F0
607A00000000410109F1C0831313232333334349F3901059C01009F34034203005F570
1305F3401015F2014434F4E544143544C4553532F4D414754454B2044DFDF4D273B353
33235303030303530303033435353D32353039303030303030303030303030303030303
03F000000D5950C1A",
      "RawData":
"AaP5ggGf399UCpAQAQuZmZkAAIDf31UBgt/fJQiZK4CIGRYHEPqCAXzwggF48QXf3xoBA
fiCAN7f31mCAMBfTZn+7SQoJXGzrJJvP5uhQrOYBUUqP3yPSRmLKYovxB4sOCxczYlP1ez
k4jKo1w5X45/7tqww6UJQybwlWbtqy7+efBhbH9FvnOLXnrc8pYQ55nvdzaivJXyeQhZc4
GmxKeU6erW6wiz0OS5OJ07EJSz1Oci0O0c7zc0dIUGdfjU81+fP+mJLrMKiP6Yz4Y1jwlh
uxVXGmLCsOIh2Sm92ef4dbOl1p+4q9aFAQLKmAU5dGNWxCfe2cvKnreHFuQrf31YKkBABC
5mZmQAAgN/fVwGA399YAQD3ggCL399AAQBfJQMgCQFfJAMlCTBfKgIIQJ8CBgAAAAABAJ8
GB6AAAAAEEBCfHAgxMTIyMzM0NJ85AQWcAQCfNANCAwBfVwEwXzQBAV8gFENPTlRBQ1RMR
VNTL01BR1RFSyBE399NJzs1MzI1MDAwMDUwMDAzNDU1PTI1MDkwMDAwMDAwMDAwMDAwMDA
wPwAAANWVDBo=",
      "KSN": "9010010B999999000080",
      "DeviceSerialNumber": "992B808819160710",
```

```
      "EncryptionType": "80",
      "NumberOfPaddedBytes": 0,
      "EMVSREDData":
"5F4D99FEED24282571B3AC926F3F9BA142B39805452A3F7C8F49198B298A2FC41E2C3
82C5CCD894FD5ECE4E232A8D70E57E39FFBB6AC30E94250C9BC2559BB6ACBBF9E7C185
B1FD16F9CE2D79EB73CA58439E67BDDCDA8AF257C9E42165CE069B129E53A7AB5BAC22
CF4392E4E274EC4252CF539C8B43B473BCDCD1D21419D7E353CD7E7CFFA624BACC2A23
FA633E18D63C2586EC555C698B0AC3888764A6F7679FE1D6CE975A7EE2AF5A14040B2A
6014E5D18D5B109F7B672F2A7ADE1C5B90A",
      "MerchantData":
"DFDF4001005F25032009015F24032509305F2A0208409F02060000000001009F0607A
00000000410109F1C0831313232333334349F3901059C01009F34034203005F5701305
F3401015F2014434F4E544143544C4553532F4D414754454B2044DFDF4D273B3533323
5303030303530303033435353D32353039303030303030303030303030303030303F"
,
      "FallbackIndicator": "",
      "MaskedICCTrack2":
"3B353332353030303035303030333435353D32353039303030303030303030303030303
03030303F",
      "ServiceCode": "",
      "CardHolderName": "434F4E544143544C4553532F4D414754454B2044",
      "CardType": "",
      "ApplicationIdentifier": "A0000000041010",
      "NumberOfPaddedBytesForBatch": 0,
      "EMVSREDDataForBatch": null
   }
]}
```


4.9 Transaction Output ARPC

Returned after RequestSendAcquirerResponse.

Example:


```
{"TransactionOutput": {
   "TransactionType": 1,
   "TransactionStatus": 0,
   "OperationStatus": 0,
   "BatchData":
"01E7F98201E3DFDF540A950003000000120071BDFDF550182DFDF250898D90C660E0
70F0EFA8201C0F08201BCF105DFDF1A0100F8820106DFDF598200E8CD1C5D6F158B03D
D94B967F26B5C5A8C925D938BE6F253B5842541EC83C36F9E36260339C8F7D3AA94212
0C21D67A5F691D1EB63E5BD51FBBDFA188D051CBF0477DFAEBCA4C388FC84ED1DC413B
0E2AF99890ADFFA87A630A81D42AD9BF08A2F69378B11F76C0111575706A3A328D7400
E9F63E7A209BEC22AD1CAEBE5952AE954E76FEB377C42920A6CCC4C42BC1B1AE773F0C
A9442B8E6B2A99F0BA52D8D665F54DC5AEC98C9F713F2D78EFC6A31A92A76AE4AE1405
A5E85EC1C0FBC9112791CCA7CC5C51478E76C4A7FEDC1FE48D406DB44114A780ACB086
259A5F409BCA61CF06419E7AAD9C3DFDF560A950003000000120071BDFDF570180DFD
F580100F78200A7DFDF4001015F25030907015F24032212315F2A0208409F020600000
00001009F0607A00000000310109F120F4352454449544F20445205649534193F1C083
1313232333334349F3901059C01009F34031E03005F5701305F3401015F201A5649534
120414351554952455220544553542F434152442030329F110101DFDF4D273B3434323
```

```
730303030303930303030303333333373D3232313230303030303030303030303030303F0
00000000000006B2E9690",
   "RawBatchData":
"Aef5ggHj399UCpUAAwAAAAEgBxvf31UBgt/fJQiY2QxmDgcPDvqCAcDwggG88QXf3xoBA
PiCAQbf31mCAOjNHF1vFYsD3ZS5Z/JrXFqMkl2Ti+byU7WEJUHsg8NvnjYmAznI99OqlCE
gwh1npfaR0etj5b1R+736GI0FHL8Ed9+uvKTDiPyE7R3EE7Dir5mJCt/6h6YwqB1CrZvwi
i9pN4sR92wBEVdXBqOjKNdADp9j56IJvsIq0crr5ZUq6VTnb+s3fEKSCmzMTEK8Gxrnc/D
KlEK45rKpnwulLY1mX1TcWuyYyfcT8teO/GoxqSp2rkrhQFpehewcD7yREnkcynzFxRR45
2xKf+3B/kjUBttEEUp4CssIYlml9Am8phzwZBnnqtnD399WCpUAAwAAAAEgBxvf31cBgN/
fWAEA94IAp9/fQAEBXyUDCQcBXyQDIhIxXyoCCECfAgYAAAAAAQCfBgegAAAAAxAQnxIPQ
1JFRElUTyBERSBWSVNBnxwIMTEyMjMzNDSfOQEFnAEAnzQDHgMAX1cBMF80AQFfIBpWSVN
BIEFDUVVJUkVSIFRFU1QvQ0FSRCAwMp8RAQHf300nOzQ0MjcwMDAwMDkwMDAwMDMzMzc9M
jIxMjAwMDAwMDAwMDAwMDA/AAAAAAAAAGsulpA=",
   "KSN": "95000300000000120071B",
   "DeviceSerialNumber": "98D90C660E070F0E",
   "EncryptionType": "80",
   "MerchantData": null
}}
```


4.10 Data Output

Returned after sending a SendCommand. The Data output contains the respsonse report sent from the device. SET returns the ACK status. GET returns the report data in response to the requested command.

Example of a SET to DynaPro:
01 – is an ACK response report.
00 – is an OK Status for the command.
0E – is the ID for the command Get Information.


```
{
   "Data":
"01000E0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000",
   "AdditionalOutputData": null
}
```


Example of a GET to DynaPro:
0E – is a Get Information respsonse report.
02 – is the Info ID for KSN.
01 – is a Key Status of OK for this key.
14 – is the Data Length in Hex.
393530303033303030303030303031323030373346 – is the Data in Hex (KSN) for this command.


```
{
   "Data":
"0E020114393530303033303030303030313230303733460000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000",
   "AdditionalOutputData": null
}
```


4.11 OperationStatus Output

Returned after RequestOperationStatus.

Example :

```
{
   "OperationStatus": 0,
   "DeviceID": null,
   "CreateNewConnectin" : false,
   "AdditionalOutputData": null
}
```

Appendix A TLV Data Format

A.1 ARQC Message Format

This section gives the format of the ARQC Message delivered in the ARQC Message notification. It is a TLV object with the following contents:

```
F9<len>/* container for MAC structure and generic data */
       DFDF54(MAC KSN)<len><val>
       DFDF55(MAC Encryption Type)<len><val>
       DFDF25(IFD Serial Number)<len><val>
       FA<len>/* container for generic data */
              70<len>/*container for ARQC */
                     DFDF53<len><value>/*fallback indicator */
                     5F20<len><value>/*cardholder name */
                     5F30<len><value>/*service code */
                     DFDF4D<len><value>/* Mask T2 ICC Data */
                     DFDF52<len><value>/* card type */
                     F8<len>/*container tag for encryption */
                            DFDF59(Encrypted Data Primitive)<len><Encrypted Data val (Decrypt
                            data to read tags)>
                            DFDF56(Encrypted Transaction Data KSN)<len><val>
                            DFDF57(Encrypted Transaction Data Encryption Type)<val>
                            DFDF58(# of bytes of padding in DFDF59)<len><val>
(Buffer if any to be a multiple of 8 bytes)
CBC-MAC (4 bytes, always set to zeroes)
```

A.2 ARQC Response (from online processing)

This section gives the format of the data for the Online Processing Result / Acquirer Response message. This request is sent to the reader in response to an ARQC Message notification from the reader. It is a TLV object with the following contents:

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

A.3 Transaction Result Message – Batch Data Format

This section gives the format of the data the device uses to do completion processing

```
FE<len> /* container for generic data */
       DFDF25(IFD Serial Number)<len><val>
       FA<len>/* container for generic data */
              F0<len> /* Transaction Results */
                     F1<len> /* container for Status Data */
                     … /* Status Data tags */

                     F2<len>/* container for Batch Data */
                     … /* Batch Data tags defined in DFDF17 */
                     …/* Note: Sensitive Data cannot be defined in DFDF17 */

                     F3<len>/* container for Reversal Data, if any */
                     … /* Reversal Data tags defined in DFDF05 */
                     …/* Note: Sensitive Data cannot be defined in DFDF05 */

                     F7<len>/* container for Merchant Data */
                     … /* < Merchant Data tags */

                     F8<len> /* container tag for encrypted data */
                            DFDF56(Encrypted Transaction Data KSN)<len><val>
                            DFDF57(Encrypted Transaction Data Encryption Type)<val>

                            FA<len> /* container for generic data */
                               DF30(Encrypted Tag 56 TLV, T1 Data)<len><val>
                               DF31(Encrypted Tag 57 TLV, T2 Data)<len><val>
                               DF32(Encrypted Tag 5A TLV, PAN)<len><val>
                               DF35(Encrypted Tag 9F1F TLV, T1 DD)<len><val>
                               DF36(Encrypted Tag 9F20 TLV, T2, DD)<len><val>
                               DF37(Encrypted Tag 9F61 TLV, T2 CVC3)<len><val>
                               DF38(Encrypted Tag 9F62 TLV, T1,PCVC3)<len><val>
                               DF39(Encrypted Tag DF812A TLV, T1 DD)<len><val>
                               DF3A(Encrypted Tag DF812B TLV), T2 DD<len><val>
                               DF3B(Encrypted Tag DFDF4A TLV, T2 ISO Format)<len><val>
```

A.4 DeviceID URI

Parameter (type)	Description

DeviceID
(string)
	URI of the device.
	
	For USB devices, use the form: an empty string to open the first device found.
	`USB://DEVICESERIALNUMBER`
	Examples,
	`"DeviceID" : ""`
	`"DeviceID" : null`
	`"DeviceID" : "USB://99261829170E0810"`
	
	For Ethernet devices, use the form:
	`IP://IPAddress:PORT`
	Example,
	`"DeviceID" : "IP://10.57.10.180:26"`
	
	For 802.11 Wireless devices, use in the form:
	`TLS12://TLSDEVICESERIALNUMBER`
	`TLS12TRUST://TLSDEVICESERIALNUMBER`
	Examples,
	`"DeviceID" : "TLS12://TLS99261829170E0810"`
	`"DeviceID" : "TLS12TRUST://TLS99261829170E0810"`
	
	For BLE devices, use the form:
	`BLEEMV://DEVICENAME`
	Example,
	`"DeviceID" : "BLEEMV://DPG123456789A"`
	DEVICENAME is listed in the operating system Bluetooth settings.