**EasyLink solution detail design**

Version 1.00.08

Release Date: Dec 19, 2016

Copyright © 2000-2016 PAX Computer Technology (Shenzhen) Co., Ltd.

All rights reserved. No part of the contents of this document may be reproduced or transmitted in any form without the written permission of PAX Computer Technology (Shenzhen) Co., Ltd. The information contained in this document is subject to change without notice. Although PAX Computer Technology (Shenzhen) Co., Ltd. has attempted to ensure the accuracy of the contents of this document, this document may include errors or omissions. The examples and sample programs are for illustration only and may not be suited for your purpose. You should verify the applicability of any example or sample program before placing the software into productive use.

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| V1.00.00 | July 13, 2016 | Idina Zhang | Initial version |
| V1.00.01 | July 20, 2016 | Idina Zhang | 1. Modified below API:  GetPinBlock,  EncryptData,  StartTransaction,  CompleteTransaction,  ShowMsgBox,  ShowInputBox,  ShowMenu,  2. Add “Amount Confirm” to mandatory UI definition;  3. Add below parameter configuration to “Terminal configuration”:  EMV AID configuration,  EMV CAPK,  EMV Revoked CAPK,  ICS,  Terminal common EMV configuration,  Paywave configuration,  Paywave AID configuration,  Sub node “Inter\_wareFlmtByTransType”,  Program ID configuration,  Paypass configuration,  Paypass AID configuration,  Other contactless EMV parameter configuration,  UI XML file configuration,  4. Modified below appendix:  Application parameter list,  Transaction parameter list,  Data to be set before StartTransaction,  Data to be set before CompleteTransaction, |
| V1.00.02 | Aug 22, 2016 | Idina Zhang | 1. Modified command list, change the first byte of command from 0x90 to 0x80;  2. Add command 0x8050, switch protocol;  2. Add TMS proxy chapter;  3. Modified description for <SetData>, <GetData> APIs;  4. Add error code definition to the appendix; |
| V1.00.03 | Aug 26, 2016 | Idina Zhang | 1. Added “EL\_PARAM\_RET\_API\_ORDER\_ERR” in Parameter management return code.  2. Added Paypass torn log block range and TLV data saving block range in Appendix 2 – Table file block distribution  3. Modified Parameter management part in Internal interface.  4. Added Appendix 3 - Terminal Information List, and added PINBlockMode in Appendix4 – Application parameter list. Modified parameters lists tag. |
| V1.00.04 | September 19, 2016 | Idina Zhang | 1. Modify Appendix3, 4, 5; |
| V1.00.05 | September 29, 2016 | Idina Zhang | 1. Modify <FileDownload> API in §3.6 File download;  2. Modify <ShowMsgBox>, <ShowInputBox>, <ShowMenu> API in §4.2 UI module; |
| V1.00.06 | November 2, 2016 | Idina Zhang | 1. Add §4.7 TMS proxy;  2. Add §4.8 File system;  3. Modify Appendix 10; |
| V1.00.07 | November 4, 2016 | Idina Zhang | 1. Modify Appendix 3, 5, 7;  2. Update communication protocol; |
| V1.00.08 | December 19, 2016 | Idina Zhang | 1. Add parameter 0x031A;  2. Add description for EncryptData; |
| V1.00.10 | March 22, 2017 | Idina Zhang | 1. Add encryption type: 0X03 MAC;  2. Add ExpressPay parameter configuration; |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table1 – revision history

Contents

[Revision history 3](#_Toc478130639)

[1 Introduction 10](#_Toc478130640)

[1.1 Purpose 10](#_Toc478130641)

[1.2 Background and goals 10](#_Toc478130642)

[1.3 Audience 10](#_Toc478130643)

[1.4 Abbreviation 10](#_Toc478130644)

[1.5 Reference 10](#_Toc478130645)

[2. Communication protocol 11](#_Toc478130647)

[2.1 Data link layer communication protocol 11](#_Toc478130648)

[2.1.1 Communication mode 11](#_Toc478130649)

[2.1.2 Communication packet format 12](#_Toc478130650)

[2.1.3 Resend mechanism 13](#_Toc478130651)

[2.1.4 Loss-Sync processing mechanism 13](#_Toc478130652)

[2.1.5 Packet size and packet split mechanism 14](#_Toc478130653)

[2.2 Message processing mechanism 14](#_Toc478130654)

[2.2.1 Data format in messaging layer 14](#_Toc478130655)

[2.2.2 Command list 15](#_Toc478130656)

[3 Interface definition for master device and POS terminal 16](#_Toc478130657)

[3.1 COMM module 16](#_Toc478130658)

[3.1.1 Connect 16](#_Toc478130659)

[3.1.2 Disconnect 16](#_Toc478130660)

[3.2 UI module 16](#_Toc478130661)

[3.2.1 ShowMsgBox 16](#_Toc478130662)

[3.3 Security module 17](#_Toc478130667)

[3.3.1 GetPinBlock 17](#_Toc478130668)

[3.3.2 EncryptData 18](#_Toc478130669)

[3.4 Transaction module 18](#_Toc478130670)

[3.4.1 StartTransaction 18](#_Toc478130671)

[3.4.2 CompleteTransaction 19](#_Toc478130679)

[3.5 Parameter management module 19](#_Toc478130680)

[3.5.1 SetData 19](#_Toc478130681)

[3.5.2 GetData 20](#_Toc478130690)

[3.6 File download 21](#_Toc478130691)

[3.7 Switch compatible mode 22](#_Toc478130692)

[3.8 TMS proxy module 23](#_Toc478130693)

[3.8.1 TMS proxy communication protocol 23](#_Toc478130694)

[3.8.2 TMS proxy communication package format 23](#_Toc478130695)

[3.8.3 TMS proxy command list 24](#_Toc478130696)

[3.8.4 GetTermSN 24](#_Toc478130697)

[3.8.5 GetTermExtSN 25](#_Toc478130698)

[3.8.6 GetTermVerInfo 25](#_Toc478130699)

[3.8.7 GetTerminalInfo 26](#_Toc478130700)

[3.8.8 SetTaskList 27](#_Toc478130701)

[3.8.9 SaveFileData 29](#_Toc478130702)

[3.8.10 GetTaskList 29](#_Toc478130703)

[4 Internal interface definition for POS terminal 30](#_Toc478130704)

[4.1 COMM module 31](#_Toc478130705)

[4.1.1 MsgInit 31](#_Toc478130706)

[4.1.2 MsgSend 31](#_Toc478130712)

[4.1.3 MsgRecv 31](#_Toc478130713)

[4.2 UI module 32](#_Toc478130714)

[4.2.1 ShowMsgBox 32](#_Toc478130715)

[4.2.2 ShowInputBox 32](#_Toc478130723)

[4.2.3 ShowMenu 32](#_Toc478130724)

[4.3 Security module 33](#_Toc478130725)

[4.3.1 GetPinBlock 33](#_Toc478130726)

[4.3.2 EncryptData 33](#_Toc478130734)

[4.4 Transaction flow 34](#_Toc478130735)

[4.4.1 StartTransaction 34](#_Toc478130736)

[4.4.2 CompleteTransaction 36](#_Toc478130745)

[4.5 Parameter management 37](#_Toc478130746)

[4.5.1 SetData 37](#_Toc478130747)

[4.5.2 GetData 38](#_Toc478130748)

[4.6 FileDownload 39](#_Toc478130757)

[4.7 TMS Proxy 39](#_Toc478130758)

[4.7.1 GetTermSN 39](#_Toc478130761)

[4.7.2 GetTermExtSN 40](#_Toc478130762)

[4.7.3 GetTermVerInfo 40](#_Toc478130763)

[4.7.4 GetTerminalInfo 40](#_Toc478130764)

[4.7.5 RemoteDownload\_TaskInfo 40](#_Toc478130765)

[4.7.6 RemoteDownload\_SaveData 41](#_Toc478130766)

[4.7.7 RemoteDownload\_TaskAsk 41](#_Toc478130767)

[4.8 File system 41](#_Toc478130778)

[5 Mandatory UI definition in POS terminal 42](#_Toc478130779)

[5.1 Prompt for card 42](#_Toc478130782)

[5.1.1 Swipe/Insert/Tap Card 42](#_Toc478130783)

[5.1.2 Fallback Swipe 42](#_Toc478130788)

[5.1.3 Tap Card Again 43](#_Toc478130789)

[5.1.4 Remove Card 44](#_Toc478130790)

[5.2 Processing 45](#_Toc478130791)

[5.3 See Phone 45](#_Toc478130792)

[5.4 Application Selection 46](#_Toc478130793)

[5.4.1 Select EMV application 46](#_Toc478130794)

[5.4.2 Select EMV application again 47](#_Toc478130802)

[5.5 PIN Entry 48](#_Toc478130803)

[5.5.1 Enter PIN 48](#_Toc478130804)

[5.5.2 Enter Offline PIN Again 49](#_Toc478130813)

[5.5.3 Last Chance to Enter PIN 49](#_Toc478130814)

[5.5.4 PIN Verify OK 50](#_Toc478130815)

[5.6 Amount Confirm 50](#_Toc478130816)

[6. Terminal configuration 52](#_Toc478130817)

[6.1 XML file data format 52](#_Toc478130818)

[6.2 EMV parameter XML file configuration 52](#_Toc478130819)

[6.2.1 EMV AID configuration 52](#_Toc478130820)

[6.2.2 EMV CAPK 53](#_Toc478130821)

[6.2.3 EMV Revoked CAPK 54](#_Toc478130822)

[6.2.4 ICS (Implementation Conformance Statement) Configuration 54](#_Toc478130823)

[6.2.5 Terminal common EMV configuration 59](#_Toc478130824)

[6.3 Contactless EMV parameter XML file configuration 60](#_Toc478130825)

[6.3.1 Paywave parameter configuration 60](#_Toc478130826)

[6.3.2 Paypass parameter configuration 65](#_Toc478130827)

[6.3.3 ExpressPay parameter configuration 69](#_Toc478130828)

[6.4 UI XML file configuration 71](#_Toc478130829)

[6.4.1 TextBox 71](#_Toc478130830)

[6.4.2 InputBox 72](#_Toc478130831)

[6.4.3 Menu 72](#_Toc478130832)

[Appendix 73](#_Toc478130833)

[Appendix 1 – Return code 73](#_Toc478130834)

[Base return code 73](#_Toc478130835)

[COMM return code 73](#_Toc478130836)

[UI return code 73](#_Toc478130837)

[Security return code 74](#_Toc478130838)

[Transaction flow return code 74](#_Toc478130839)

[Parameter management return code 75](#_Toc478130840)

[TMS Proxy return code 75](#_Toc478130841)

[FileDownload return code 75](#_Toc478130842)

[Common return code 76](#_Toc478130843)

[Appendix 2 – Table file block distribution 76](#_Toc478130844)

[Appendix 3 – Terminal Information List 77](#_Toc478130845)

[Appendix 4 – Application parameter list 80](#_Toc478130846)

[Appendix 5 – Transaction parameter list 81](#_Toc478130847)

[Appendix 6 – Data to be set before StartTransaction 83](#_Toc478130848)

[Appendix 7 – Data to be set before CompleteTransaction 83](#_Toc478130849)

[Appendix 8 - EMV TAGs 84](#_Toc478130850)

[EMV TAGs can be accessed after StartTransaction: 84](#_Toc478130851)

[CLSS TAGs can be accessed after StartTransaction 86](#_Toc478130852)

[EMV TAGs can be accessed after CompleteTransaction 86](#_Toc478130853)

[CLSS TAGs can be accessed after CompleteTransaction 86](#_Toc478130854)

[Appendix 9 – Value attribute 86](#_Toc478130855)

[Appendix 10 – File type description for file download 87](#_Toc478130856)

# Introduction

## Purpose

This document is detail design of EasyLink solution. It mainly describe the detail design and protocol between the mPOS and Master device which give reference when coding.

## Background and goals

The original MPOS solution provides a set of commands with slight granularity which make it difficult for application in master device (like Android/IOS Windows device) to use. So the purpose of this specification for EasyLink solution is to design an optimization program structure with better extendibility and provide a set of commands with larger granularity to improve the interaction efficiency and allow quick and easy integration with customized applications in master device.

## Audience

All the programmers and customers who need to develop the mPOS terminal and master device.

## Abbreviation

|  |  |
| --- | --- |
| Name | Description |
| **POS** | Point of sale |
| **PDK** | PAX Platform Development Kit |
| **SDK** | Software development kit |
| **RKI** | Remote key injection |
| **KMS** | Key management system |
| **PAN** | Primary account number |
| **TM** | Terminal management |
| **TMS** | Terminal management systme |
| **AID** | Application identifier |
| **CAPK** | Certification authority public key |

Table2 - Abbreviation

## Reference

PAX\_PDK\_API\_Programming\_Guide\_V1.00.00.doc

EUI详细设计说明(V1.1.01).docx

4.3book3\_Application.pdf

PAX EMV Kernel API Programming Guide.pdf

D180与上位机通讯链路层协议.docx

EasyLink outline design specification\_v1.00.02\_20160701173030.docx



# Communication protocol

## Data link layer communication protocol

### Communication mode

This communication mode is applied to the request and response between the POS terminal and master device: the master device send request packet, then POS terminal receive and process the request command, and send the response packet back to the master device.

For each package, the receiver needs to respond an ACK (0x06) if received successfully, otherwise it needs to respond a NAK (0x15). If the sender doesn’t receive an ACK or NAK in 150ms after it sends one package, it will retry 3 times

If the length of transmission data exceeds the maximum length of frame, then send the transmission data in multi frame. If the ETX is 0x17, it indicates there exists subsequent frames.

See the chart below:

* Single frame

|  |  |  |
| --- | --- | --- |
| Sender (master device) |  | Receiver（POS terminal） |
| Request package | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **ACK** |
|  | 🡨🡨🡨 | **Response package** |
| ACK | 🡪🡪🡪 |  |

Table3 – single frame

* Multi frames

|  |  |  |
| --- | --- | --- |
| Sender (master device) |  | Receiver (POS terminal) |
| 1st frame of request package | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **ACK** |
| 2nd frame of request package | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **ACK** |
| nth frame of request package | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **ACK** |
|  | 🡨🡨🡨 | **1st frame of response package** |
| ACK | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **2nd frame of response package** |
| ACK | 🡪🡪🡪 |  |
|  | 🡨🡨🡨 | **nth frame of response package** |
| ACK | 🡪🡪🡪 |  |

Table4 – multi frame

### Communication packet format

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | | **Frame No.** | | **Data Length** | | **Data** | **LRC** | **ETX** |
| PaNoH | PaNoL | FrNoH | FrNoL | LenH | LenL |  |
| 0x02  (1 byte) | Version of package, fixed to be 0x01 for now | PaNoH=PaNo/256  PaNoL=PaNo%256  Start from 1000 | | PaNoH=PaNo/256  PaNoL=PaNo%256  Sstetart from 1 | | 2 byte data domain length used to respresent the data domain length. The value is 0, indicating no data domain of 2 byte length, the former byte is data field length Len/256 and the latter byte is Len%256. In protocol, all involving length or size data field are the high byte at the frontt, low byte in the post. | |  | LRC is the data XOR start from starting the character. (1 byte) | EOF：  0x03 Last frame  0x17 there exits subsequent frame |

Table5 – communication packet format

#### Packet No.

Packet No. is maintained by master device which is used to identify the request packet. It starts from 1000, while the frame No. below to 1000 is used to identify the control frame, such as acknowledge frame ACK/NAK, synchronizing frame, and so on.

For each request packet, the master device shall increase the packet No. by 1. Meanwhile, the POS terminal shall use the same packet No. in its response packet.

If the packet No. between the receiver and sender gets out of synchronization, the master device shall send the synchronizing frame first and then send the request packet.

#### Frame No.

Frame No. is used to identify the sequence in multi frame communication. It starts from 1, and maintained by the sender. The frame No. should be increased by 1 by the sender in multi frame communication.

#### ACK frame

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | | **Frame No.** | | **Data Length** | | **Data** | **LRC** | **ETX** |
| PaNoH | PaNoL | FrNoH | FrNoL | LenH | LenL |  |
| 0x02 | 0x01 | 0x00,0x00 | |  | | 0x00,0x01 | | 0X06 | LRC | 0x03 |

Table6 – ACK packet format

#### NAK frame

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | | **Frame No.** | | **Data Length** | | **Data** | **LRC** | **ETX** |
| PaNoH | PaNoL | FrNoH | FrNoL | LenH | LenL |  |
| 0x02 | 0x01 | 0x00,0x00 | |  | | 0x00,0x01 | | 0X15 | LRC | 0x03 |

Table7 – NAK packet format

#### Synchronizing/Handshake frame

|  |  |  |
| --- | --- | --- |
| Sender (master device) |  | Receiver (terminal) |
| Synchronizing request frame | 🡪🡪🡪 |  |
|  |
|  | 🡨🡨🡨 | Synchronizing response frame |

Table8 – Synchronizing/Handshake frame

See the below request frame of synchronizing frame:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | | **Frame No.** | | **Data Length** | | **Data** | **LRC** | **ETX** |
| PaNoH | PaNoL | FrNoH | FrNoL | LenH | LenL |  |
| 0x02 | 0x01 | 0x00,0x01 | | 0x00，0x01 | | 0x00,0x00 | | N/A | LRC | 0x03 |

Table9–request frame of synchronizing frame format

See the below response frame of synchronizing frame:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | | **Frame No.** | | **Data Length** | | **Data** | **LRC** | **ETX** |
| PaNoH | PaNoL | FrNoH | FrNoL | LenH | LenL |  |
| 0x02 | 0x01 | 0x00,0x01 | | 0x00，0x01 | | 0x00,0x08 | | PaSize+ FrSize | LRC | 0x03 |

Table10–response frame of synchronizing frame format

Packet size: the maximum length of POS terminal supported, 4 bytes, Big Endian.

Frame size: the maximum length of POS terminal supported, 4 bytes, Big Endian.

### Resend mechanism

Resending will be happened in the below situations:

* NAK retransmission: the sender shall retransmit the packet after receiving NAK;
* Timeout retransmission: then sender shall retransmit the packet if it doesn’t receive the ACK within 150ms;

If the receiver receive resent packet (the received frame No. is less than the expected frame No.), then receiver would discard the current packet.

If the sender still get fail after resending the packet, then it shall return an error.

### Loss-Sync processing mechanism

Loss-Sync will be happened in the below situation:

* For master device, if the packet No. in request packet is not equal to the packet No. in response packet;
* For POS terminal, if the packet No. in request packet is no equal to the expected packet No., or the frame No. in request packet is no is no equal to the expected frame No. and is not one interval less than expected frame No.

The master device shall be the subject to control the loss-sync processing. Once the master device detects loss-sync, it shall send the synchronizing packet to the POS terminal. And then the POS terminal should reset to synchronize the packet No. and send back the corresponding response.

### Packet size and packet split mechanism

Packet size and packet split mechanism (frame size) shall be synchronized by synchronizing request packet which is sent by the POS terminal. The packet size should be the equal to maximum length of frame in the POS terminal.

Due to the memory limitation of D180S, the size of frame in D180S shall be 1000 bytes in Bluetooth and USB communication.

## Message processing mechanism

The messaging protocol is responsible for command parsing and delivery to the business layer for further processing or display.

### Data format in messaging layer

The below table shows the request data format:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | **Frame No.** | **Data Length** | **Data** | | **LRC** | **ETX** |
| Command | Data |
| 0x02  1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes | 2 bytes | The data content of communication needs. The definition is parsed by each command and some command may not have this field. | 1 byte | 1 byte |
| Data link layer protocol | | | | | Messaging layer | | Data link layer protocol | |

Table11–request data format in messaging layer

The below table shows the response data format:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Version** | **Packet No.** | **Frame No.** | **Data Length** | **Data** | | | | **LRC** | **ETX** |
| Command | Status code | Response MSG | Data |
| 0x02  1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes | 2 bytes | 4 bytes | 64 bytes | The data content of communication needs. The definition is parsed by each command and some command may not have this field. | 1 byte | 1 byte |
| Data link layer protocol | | | | | Messaging layer | | | | Data link layer protocol | |

Table12–response data format in messaging layer

### Command list

Command list between master device and POS terminal:

|  |  |  |  |
| --- | --- | --- | --- |
| Command | | Description | Other |
| Command type | Command code |
| 0x80 | 0x00 | Connect | CMD\_COMM\_CONNECT |
| 0x80 | 0x01 | Disconnect | CMD\_COMM\_DISCONNECT |
|  |  |  |  |
| 0x80 | 0x10 | ShowMsgBox | CMD\_UI\_SHOW\_MSG\_BOX |
|  |  |  |  |
| 0x80 | 0x20 | GetPinBlock | CMD\_GET\_PIN\_BLOCK |
| 0x80 | 0x21 | EncryptData | CMD\_ENCRYPT\_DATA |
|  |  |  |  |
| 0x80 | 0x30 | StartTransaction | CMD\_TRANS\_START |
| 0x80 | 0x31 | CompleteTransaction | CMD\_TRANS\_COMPLETE |
|  |  |  |  |
| 0x80 | 0x40 | SetData | CMD\_PARAM\_SET\_DATA |
| 0x80 | 0x41 | GetData | CMD\_PARAM\_GET\_DATA |
|  |  |  |  |
| 0x80 | 0x50 | FileDownload | CMD\_FILE\_DOWNLOAD |
|  |  |  |  |
| 0x80 | 0x60 | Switch compatible communication mode | CMD\_SWITCH\_COMPATIBLE\_COMM\_MODE |
|  |  |  |  |

Table13–command list

# Interface definition for master device and POS terminal

## COMM module

### Connect

**Description:**

This API is used to create connection between the POS and Android/IOS/Windows device.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x00 | N/A |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x00 | As follows | N/A |

**Return code:**

See appendix [COMM return code](#_COMM_return_code)

### Disconnect

**Description:**

This API is used to drop connection between the POS and Android/IOS/Windows device.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x01 | N/A |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x01 | As follows | N/A |

**Return code:**

See appendix [COMM return code](#_COMM_return_code)

## UI module

### ShowMsgBox

**Description:**

The terminal provides this API to Android/IOS/Windows device to fulfill all the UI related processing with the UI XML files, like show some message, pictures, and etc.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x10 | See request data field |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | PageNameLen | Length of PageName |
| 0x0001 | N | PageName | PageName |
| 0x0001 + N | 2 | Timeout |  |
| 0x0001 + N + 2 | 1 | WidgetNameLen[1] | Length of WidgetName |
| 0x0001 + N + 3 | N1 | WidgetName[1] | WidgetName |
| 0x0001 + N + 3 + N1 | 1 | TextLen[1] | Length of Text |
| 0x0001 + N + 3 + N1 + 1 | N2 | Text[1] | Text string: |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x10 | As follows | N/A |

**Return code:**

See appendix [UI return code](#_UI_return_code)



## Security module

### GetPinBlock

**Description:**

This API is used to get PIN block.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x20 | See request data field |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | PANLen | Length of PAN |
| 0x0001 | N | PAN | PAN |
|  |  |  |  |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x10 | As follows | See PIN Block table |

**PIN Block table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 8 | PIN Block | PIN Block |
| 0x0008 | 10 | KSN | KSN |

**Return code:**

See appendix [Security return code](#_Security_return_code)

### EncryptData

**Description:**

This API is used to encrypt data.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x21 | See request data field |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | DataLen | Length of data;  Note:  1. For RSA encryption, the length of data to be encrypted should be equal to modulus of RSA key.  2. For TDES encryption, the length of the data to be encrypted should be divisible by 8, if not, the terminal pad 0x00 to the right.  3. For MAC encryption, the length of the data that used to do calculation should be less than 1024 bytes; if the length is not multiple by 8, 0x00 will be padded automatically. |
| 0x0001 | N | Data | Data |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x21 | As follows | See Encryption Block table |

**Encryption Block table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | EncryptionBlockLen | Length of encryption block |
| 0x0002 | N | EncryptionBlock | Encryption block |

**Return code:**

See appendix [Security return code](#_Security_return_code)

## Transaction module

### StartTransaction

**Description:**

This API is used to detect card, then do the corresponding processing according to the card type, if MSR, then read MSR data, if contact EMV chip, then do the contact EMV processing (all the offline processing) according to the contact EMV specification, if contactless EMV card, then do the corresponding contactless EMV processing (all the offline processing).

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x30 | N/A |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x30 | As follows | N/A |

**Return code:**

See appendix [Transaction flow return code](#_Transaction_flow_return)

**Note:**

Before do transaction in StartTransaction, there’re some mandatory transaction data should be set into POS terminal, see [Appendix 5 – Data to be set before AuthorizeCard](#_Appendix_5_–)



### CompleteTransaction

**Description:**

If online authentication is needed for contact EMV or contactless EMV transaction, then Android/IOS/Windows device shall call this API to complete contact EMV or contactless EMV transaction after online processing.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x31 | N/A |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x31 | As follows | N/A |

**Return code:**

See appendix [Transaction flow return code](#_Transaction_flow_return)

**Note:**

Before do transaction in CompleteTransaction, there’re some mandatory data should be set into POS terminal, see [Appendix 6 – Data to be set before CompleteOnlineTxn](#_Appendix_6_–)

## Parameter management module

### SetData

**Description:**

This API is used to set value for contact EMV or contactless EMV TAGs, transaction parameters, terminal parameters, and so on.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x40 | See request data table |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | DataType | Data type:  1: Transaction data;  2: Configuration data; |
| 0x0001 | 2 | DataLen | Length of data |
| 0x0003 | N | Data | All the data shall be expanded to double length. Allow to set tags with TLV list.  e.g. Cardholder Name (“JACK”)  “\x5F\x20\x04\x4A\x41\x43\x4B”  For Configuration Data, Please refer to Configuration Data List. |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x40 | As follows | See the response data field table |

**Response data field table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | SuccessTagListLen | Length of tag list which set success |
| 0x0002 | N | SuccessTagList | List of tags which have been set success.  All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “9F269F274F” in BCD format. |

**Return code:**

See appendix [Parameter management return code](#_Parameter_management_return)



### GetData

**Description:**

This API is used to get value for contact EMV or contactless EMV TAGs, transaction parameters, terminal parameters, and so on.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x41 | See request data table |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | DataType | Data type:  1: Transaction data;  2: Configuration data; |
| 0x0001 | 2 | DataLen | Length of data |
| 0x0003 | N | Data | List of requested tags. All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “9F269F274F” in BCD format |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x41 | As follows | See the response data field table |

**Response data field table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | SuccessTagListLen | Length of tag list which set success |
| 0x0002 | N | SuccessTagList | List of tags which have been gotten success.  All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “9F269F274F” in BCD format. |

**Return code:**

See appendix [Parameter management return code](#_Parameter_management_return)

## File download

**Description:**

This command is used to download file from Android/IOS side.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x50 | See **Request data field table** |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | FileNameLen | Length of file name |
| 0x0002 | N…32 | FileName | File name, please pay attention to the suffix of the file, see [Appendix 9 – File type description for file download](#_Appendix_9_–) |
| 0x0002 + N | 4 | FileSize | File size |
| 0x0002 + N +4 | 4 | FileOffset | File offset |
| 0x0002 + N + 4 | 4 | CurTransferLen | Current transmission length of file data, maximum transfer length shall be 2048 bytes. |
| 0x0002 + N + 2 + 2 | N1 | FileData | File data |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x50 | As follows | N/A |

**Return code:**

See appendix [Common return code](#_Common_return_code)

## Switch compatible mode

**Description:**

Since the MPOS protims uses different communication protocol compare to the protocol introduced in chapter §2. Communication protocol, therefore, this command is designed to open compatible communication mode.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x80 | 0x60 | See **Request data field table** |

**Request data field:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | ProtocolType | Protocol type:  0: multi-frame protocol which is introduced in §2 Communication protocol; (default)  1: simple protocol whose request format is: STX + CMD(2bytes) + LENGTH(2bytes) + DATA + LRC, see chapter $3.8.1 TMS proxy communication protocol; |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x80 | 0x60 | As follows | N/A |

**Return code:**

See appendix [COMM return code](#_COMM_return_code)

## TMS proxy module

The Android/IOS/Windows device download all the files from TMS server, then push all the files into terminal. So terminal uses this API to download parameter files, including EMV parameter file, EMV contactless parameter file, UI XML files, font, application, application parameter files, etc.

### TMS proxy communication protocol

Using ECI Communication Interface mode: for each communication packet, after it is receivedcorrectly, receiving end requires an immediate response ACK (0x06) characters, or response NAK (0x15). If the sender does’t receive ACK or NAK within 2000ms after sending a packet ,you need to try again four times.

See the picture below：



### TMS proxy communication package format

Data Package Format as following:

Table 2-1Package Format

**Request Data Package From Android：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STX** | **Command** | | **Data Length** | **Data** | **LRC** |
| **Module Command** | **Function Command** |
| 0x02  (1 Byte) | 1 byte | 1 byte | 2 byte data domain length used to represent the data domain length. The value is 0, indicating no data domain of 2 byte length, the former byte is data field length Len/256 and the latter byte is Len%256. In protocol, all involving length or size data field are the high byte at the front, low byte in the post. | The data content of communication needs (N byte), the field definition is parsed by each command and some command may not have this field. | LRC is the data XOR except starting the character. (1 byte) |

**Response database：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STX** | **Command** | | **Data Length** | **Data** | | **LRC** |
| **Module Command** | **Function Command** | **Return Code** | **Data Contents** |
| 0x02  (1 Byte) | 1 byte | 1 byte | 2 byte data domain length used to respresent the data domain length. The min valuer is 4, indicating no data contents, the former byte is data field length Len/256 and the latter byte is Len%256. In protocol, all involving length or size data domain are the high byte at the front, low byte in the post. | 4 byte return code is high byte at the front, int type. | The data contents of communication needs (N byte), the field definition is parsed by each command and some command may not have this field. | LRC is the data XOR except the starting character (1 byte). |

Note: Since the communication protocol of TMS proxy is different with the EasyLink communication protocol, so before use TMS proxy to download file, please call CMD\_OPEN\_COMPATIBLE\_COMM\_MODE command to open compatible communication mode.

### TMS proxy command list

Command list of TMS proxy between master device and POS terminal:

|  |  |  |  |
| --- | --- | --- | --- |
| Command | | Description | Other |
| Command type | Command code |
| 0x90 | 0x34 | GetTermSN | CMD\_TMS\_PROXY\_GET\_TERM\_SN |
| 0x90 | 0x35 | GetExtSN | CMD\_TMS\_PROXY\_GET\_EXT\_SN |
| 0x90 | 0x3d | GetVerInfo | CMD\_TMS\_PROXY\_GET\_VERSION |
| 0x90 | 0x3e | GetTerminalInfo | CMD\_TMS\_PROXY\_GET\_TERM\_INFO |
|  |  |  |  |
| 0x91 | 0x41 | SetTaskList | CMD\_TMS\_PROXY\_SET\_TASK\_LIST |
| 0x91 | 0x42 | SaveFileData | CMD\_TMS\_PROXY\_SAVE\_FILE\_DATA |
| 0x91 | 0x43 | GetTaskList | CMD\_TMS\_PROXY\_GET\_TASK\_LIST |
|  |  |  |  |

### GetTermSN

**Description**: Read the serial number of the terminal.

**Request**:

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Data Field |
| 0x90 | 0x34 | 0x00,0x00 | N/A |

**Response**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Return Code | Data Field |
| 0x90 | 0x34 | 0xNN,0xNN | as follows | SN Info Table |

**SN Info Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Length(byte) | name | Value,Description |
| 0x0000 | 1 | SN\_LEN | Length of SN |
| 0x0001 | N | SN | Serial number |

**Return Code:**

See [Base Return Code](#_Base_Return_Code:)

### GetTermExtSN

**Description**: Read the extended serial number of the terminal (for some special application requirement).

**Request**:

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Data Field |
| 0x90 | 0x35 | 0x00,0x00 | N/A |

**Response**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Return Code | Data Field |
| 0x90 | 0x35 | 0xNN,0xNN | as follows | EXSN Table |

**EXSN Info Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Length(byte) | name | Value,Description |
| 0x0000 | 1 | EXSN\_LEN | Length of EXSN |
| 0x0001 | N | EXSN | Extended Serial number |

**Return Code**:

See [Base Return Code](#_Base_Return_Code:)

### GetTermVerInfo

**Description**: Get version information of the terminal.

**Request**:

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Data Field |
| 0x90 | 0x3d | 0x00,0x00 | N/A |

**Response**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Return Code | Data Field |
| 0x90 | 0x3d | 0x00,0x08 | as follows | Version Info Table |

**Version Info Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Length | name | Value,Description |
| 0x0000 | 8 | VerInfo | Version Information(8 Bytes)  [0]=BOOT Version (ascending from 1)  [1]=Monitor Major version (ascending from 1)  [2]=Monitor minor version (ascending from 0)  [3]=Main PCB HW version: 00-31 (%d) (refer to HW version)  [4]=Interface PCB configuration : 00-31 (%d)  [5]=extended PCB configuration [6]=MSR PCB configuration [7]=Reserved |

**Return Code**:

See [Base Return Code](#_Base_Return_Code:)

### GetTerminalInfo

**Description**: Get terminal model and configuration information, information buffer should be not less than 30 bytes.

**Request**:

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Data Field |
| 0x90 | 0x3e | 0x00,0x00 | N/A |

**Response**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command Type | Command Code | Data Length | Return Code | Data Field |
| 0x90 | 0x3e | 0x00,0x1E | as follows | Terminal Info Table |

**Terminal Info Table**:

|  |  |  |  |
| --- | --- | --- | --- |
| Offset | Length | name | Value,Description |
| 0x0000 | 1 | Out\_Info[0] | Terminal model  23 - D180 |
| 0x0001 | 1 | Out\_Info[1] | Printer type |
| 0x0002 | 1 | Out\_Info[2] | MODEM configuration |
| 0x0003 | 1 | Out\_Info[3] | MODEM the highest sync baud rate |
| 0x0004 | 1 | Out\_Info[4] | MODEM the highest async. baud rate |
| 0x0005 | 1 | Out\_Info[5] | PCI configuration |
| 0x0006 | 1 | Out\_Info[6] | USB host configuration |
| 0x0007 | 1 | Out\_Info[7] | USB device configuration |
| 0x0008 | 1 | Out\_Info[8] | LAN(TCP/IP)configuration |
| 0x0009 | 1 | Out\_Info[9] | GPRS configuration |
| 0x000A | 1 | Out\_Info[10] | CDMA configuration |
| 0x000B | 1 | Out\_Info[11] | WiFi configuration |
| 0x000C | 1 | Out\_Info[12] | Contactless Card Reader configuration  0 - No contactless reader module Others - Contactless reader module |
| 0x000D | 1 | Out\_Info[13] | If has Chinese font lib  0 -not Chinese font library.  1 -Chinese font library. |
| 0x000E | 1 | Out\_Info[14] | Font lib version  0 -no font library file.  Others-the version number of fontlibrary. |
| 0x000F | 1 | Out\_Info[15] | ICC Reader configuration  0x00 -no ICC reader.  Others-has ICC reader. |
| 0x0010 | 1 | Out\_Info[16] | MSR configuration  0x00 -no MSR reader.  Others-has MSR reader |
| 0x0011 | 1 | Out\_Info[17] | Tilt Sensor configuration |
| 0x0012 | 1 | Out\_Info[18] | WCDMA configuration |
| 0x0013 | 1 | Out\_Info[19-29] | Reversed |

**Return Code**:

See [Base Return Code](#_Base_Return_Code:)

### SetTaskList

**Description:**

Set task list information. Work in conjunction with SaveFileData command to implement remote file upgrade.

The task list information includes task number and file information of each task. The file information of each task includes: file type, file No., file name, application name, application version, and force update flag.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x91 | 0x41 | See task list data |

**Task list data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | TaskNum | Task number, hex format; |
| 0x0002 | 2 | CurTaskNum | Current task number, hex format; |
| 0x0004 | 1 | FileType | File type:  0x00: download monitor  0x01: create file system  0x02: download font  0x03: delete application  0x04: download application  0x05: download parameter file  0x06: delete parameter file  0x07: delete all application  0x09: wireless driver  0x0A: download dynamic library  0x0B: delete dynamic library |
| 0x0005 | 1 | FileNo | File No. |
| 0x0006 | 2 | FileNameLen | Length of file name |
| 0x0008 | N (n…32) | FileName | File name |
| 0x0008 + N | 2 | AppNameLen | Length of application name |
| 0x0008 + N + 2 | N1 (n…32) | AppName | Application name |
| 0x0008 + N + 2 + N1 | 2 | VersionLen | Length of version |
| 0x0008 + N + 2 + N1 + 2 | N2 (n…20) | Version | Version |
| 0x0008 + N + 2 + N1 + 2 + N2 | 1 | ForceUpdateFlag | Force update flag |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x91 | 0x41 | As follows | N/A |

**Return code:**

See appendix [TMS Proxy return code](#_TMS_Proxy_return)

Attention: if the task list number is over than 10, then the task list data shall be sent in multi-package.

### SaveFileData

**Description:**

Save file content. Work in conjunction with SaveFileData command to implement remote file upgrade.

This command shall be called after <SetTaskList> command.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x91 | 0x42 | See request data table |

**Request data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 1 | FileNo. | File No. |
| 0x0001 | 4 | FileSize | File size of each file |
| 0x0005 | 4 | Offset | Offset of file content |
| 0x0009 | 4 | CurSize | Current package size |
| 0x000d | N | FileContent | File content |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x91 | 0x42 | As follows | N/A |

**Return code:**

See appendix [TMS Proxy return code](#_TMS_Proxy_return)

### GetTaskList

**Description:**

Get task list information. This command is called by Android/IOS app to get task list from terminal, then compare with the task list information in Android/IOS app and check which file need to be downloaded.

The task list information includes task number and file information of each task. The file information of each task includes: file type, file No., file name, application name, application version, force update flag.

**Request:**

|  |  |  |
| --- | --- | --- |
| Command type | Command Code | Data Field |
| 0x91 | 0x43 | See request data |

**Request data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Length(byte)** | **name** | **Value,Description** |
| 0x0000 | 2 | TaskNum | Task number, hex format; |
| 0x0002 | 2 | CurTaskNum | Current task number, hex format; |
| 0x0004 | 1 | FileType | File type:  0x00: download monitor  0x01: create file system  0x02: download font  0x03: delete application  0x04: download application  0x05: download parameter file  0x06: delete parameter file  0x07: delete all application  0x09: wireless driver  0x0A: download dynamic library  0x0B: delete dynamic library |
| 0x0005 | 1 | FileNo | File No. |
| 0x0006 | 2 | FileNameLen | Length of file name |
| 0x0008 | N (n…32) | FileName | File name |
| 0x0008 + N | 2 | AppNameLen | Length of application name |
| 0x0008 + N + 2 | N1 (n…32) | AppName | Application name |
| 0x0008 + N + 2 + N1 | 2 | VersionLen | Length of version |
| 0x0008 + N + 2 + N1 + 2 | N2 (n…20) | Version | Version |
| 0x0008 + N + 2 + N1 + 2 + N2 | 1 | ForceUpdateFlag | Force update flag |

**Response:**

|  |  |  |  |
| --- | --- | --- | --- |
| Command Type | Command Code | Return Code | Data Field |
| 0x91 | 0x41 | As follows | N/A |

**Return code:**

See appendix [TMS Proxy return code](#_TMS_Proxy_return)

Attention: if the task list number is over than 10, then the task list data shall be sent in multi-package.

# Internal interface definition for POS terminal

Below are the internal interfaces for application layer to use.

## COMM module

### MsgInit

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int iMsgInit(**const **int iPort,** const **int iPaSize,** const **int iFrSize);** | |
| **Function** | Initialize the COMM module | |
| **Parameters** | iPort [Input] | COMM port |
| iPaSize [Input] | Packet size to be set |
| iFrSize [Input] | Frame size to be set |
| **Return** |  |  |
| **Instruction** |  | |



### MsgSend

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int iMsgSend(**const **int iMsgDataLen,** const **uchar\*pucMsgData,** const **ulong ulTimeOutMs)；** | |
| **Function** | Use specified communication port to send data. | |
| **Parameters** | iMsgDataLen [Input] | Length of data to be sent |
| \*pucMsgData [Input] | Data to be sent |
| ulTimeOutMs [Input] | Timeout for receiving ACK/NAK frame (ms); |
| **Return** |  |  |
| **Instruction** |  | |

### MsgRecv

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int iMsgRcv(**const **uint uiMsgDataBufLen,** const **uchar\*pucMsgDataBuf,** const **ulong ulTimeOutMs);** | |
| **Function** | Use specified communication port to receive data. | |
| **Parameters** | uiMsgDataBufLen [Input] | Length of receiving buffer (pucMsgDataBuf) |
| \* pucMsgDataBuf [Input] | Receiving buffer |
| ulTimeOutMs [Input] | Timeout for receiving data (ms); |
| **Return** |  |  |
| **Instruction** |  | |

## UI module

### ShowMsgBox

|  |  |  |
| --- | --- | --- |
| **Prototype** | int ShowMsgBox(const uchar \*pucPageName,const ushort usMsgLength, const uchar \*pucMsg, const ushort usTimeout); | |
| **Function** | This API is used to show some message, pictures, and etc. | |
| **Parameters** | pucPageName[Input] | Page name |
| usMsgLength | Msg package length |
| \*pucMsg[Input] | Widget name msgtoshow |
|  | usTimeout[input] | Timeout |
| **Return** |  |  |
| **Instruction** |  | |



### 4.2.2 ShowInputBox

|  |  |  |
| --- | --- | --- |
| **Prototype** | int ShowInputBox(const uchar \*pucPageName,const ushort usMsgLength, const uchar \*pucMsg, uchar \*pucResponse, const ushort usTimeout); | |
| **Function** | This API is used to show InputBoxto get input data. | |
| **Parameters** | \*pucPageName [input] | Page name |
| usMsgLength[Input] | Msg package length |
| pucMsg[Input] | Widget name and text to show |
| pucResponse[output] | The data that user input |
|  | usTimeout[input] | Timeout |
| **Return** |  |  |
| **Instruction** |  | |

### 4.2.3 ShowMenu

|  |  |  |
| --- | --- | --- |
| **Prototype** | int ShowMenuList(const uchar \*pucPageName,const ushort usMsgLength, const uchar \*pucMsg, uchar \*pucResponse,const ushort usTimeout); | |
| **Function** | Show menu | |
| **Parameters** | \*pucPageName [input] | Page name |
| usMsgLength[input] | Msg package length |
| pucMsg[input] | Widget name and menu item to show |
|  | \*pucResponse[output] | Output the selected menuitem No. |
|  | usTimeout[input] | Timeout |
| **Return** |  |  |
| **Instruction** |  | |

## Security module

### GetPinBlock

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int GetPinBlock(uchar \*pucCardNo, uchar \*pucPinBlockOut, uchar \*pucKsnOut)** | |
| **Function** | Input PIN in PED, and use specific key to encrypt plaintext PIN Block, output the cipher PIN Block | |
| **Parameters** | \*pucCardNo [Input] | Primary account number |
| \*pucPinBlockOut [Output] | 8 bytes PIN Block |
| \*pucKsnOut | Current KSN |
| **Return** |  |  |
| **Instruction** |  | |



### EncryptData

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int EncryptData(uchar \*pucDataIn, uint uiDataLen, uchar \* pucEncDataOut, uint \*puiEncDataLen)** | |
| **Function** | Use specific key to encrypt data, output encryption block | |
| **Parameters** | \*pucDataIn [Input] | Data to be encrypted |
| uiDataLen [Input] | Length of data to be encrypted |
| \*pucEncDataOut [Output] | Encrypted data |
| \*puiEncDataLen [Output] | Length of encrypted data |
| **Return** |  |  |
| **Instruction** |  | |

## Transaction flow

### StartTransaction

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int StartTransaction();** | |
| **Function** | This API is used to detect card, then do the corresponding processing according to the card type, if MSR, then read MSR data, if contact EMV chip, then do the contact EMV processing (all the offline processing) according to the contact EMV specification, if contactless EMV card, then do the corresponding contactless EMV processing (all the offline processing). | |
| **Parameters** | None |  |
| **Return** |  |  |
| **Instruction** | In this API, terminal will do the following procedures according to the EMV contact specification and EMV contactless specification, and also read data from MSR:   1. DetectCard, including MSR, EMV contact chip, EMV contactless card; 2. If EMV chip card inserted; 3. Go EMV contact transaction flow; 4. App Selection 5. App Initialization 6. Get Processing Options 7. Check if need fallback, if yes, then return error, if no, then return EMV data 8. Process Restrictions 9. Process CVM 10. Offline Data Authentication 11. Terminal Risk Management 12. First GAC 13. If transaction approved or declined, remove card 14. If EMV contatctless card tapped; 15. Go EMV contactless transaction flow; 16. App Selection 17. Get Final Select Data 18. Paywave/Paypass/AE(or other CLSS kernel) processing 19. Restriction Processing 20. Offline Data Authentication 21. CVM Processing 22. Remove card 23. Outcome Check, if need tap card again, go back to step 1, if not, return the result back to Android/IOS/Windows device. 24. If MSR swiped; 25. Read MSR data; | |



Processing flow for StartTransaction:



### CompleteTransaction

|  |  |  |
| --- | --- | --- |
| **Prototype** | **Int CompleteTransaction();** | |
| **Function** | If online authentication is needed for EMV or CLSS transaction, then Android/IOS/Windows device shall call this API to complete EMV/CLSS traction after online processing, below are the procedures in this API: | |
| **Parameters** | None |  |
| **Return** |  |  |
| **Instruction** | Below are the procedures in this API:   1. if EMV chip card 2. External authentications if exists 3. Issuer script processing if exists 4. Second GAC if online authenticated 5. Remove card 6. If CLSS card 7. External Authentication 8. Issuer Script Process | |

Processing flow for CompleteTransaction:



## Parameter management

### SetData

|  |  |  |
| --- | --- | --- |
| Prototype | int SetData(unsigned char ucDataType, const unsigned char \*pucTlvList, unsigned int uiTLVListLen,  unsigned int uiSuccessTagListOutBufLen, unsigned char \*pucSuccessTagListOut, unsigned int \*puiSuccessTagListLenOut); | |
| Function | This API is use to set value for EMV/CLSS TAGs, transaction parameters, terminal parameters, and so on; | |
| Parameters | ucDataType [Input] | Data Type:  1: EMV Data.  2: Customize Data. See Appendix 3 and Appendix 4. |
| \* pucTlvList [Input] | Note: All the data shall be expanded to double length. Allow to set tags with TLV list.  e.g. Cardholder Name (“JACK”)  “\x5F\x20\x04\x4A\x41\x43\x4B”  For Configuration Data, Please refer to Configuration Data List. |
| uiTLVListLen[Input] | The length of the TLV list. |
| uiSuccessTagListOutBufLen[Input] | The length of the successful tag list output buffer. |
| \*pucSuccessTagListOut [Output] | List out TAGs which have been set success.  All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “\x9F\x26\x9F\x27\x4F” |
| \*puiSuccessTagListLenOut  [Output] | Length of successful TAGs list |
| Return |  |  |
| Instruction |  | |

### GetData

|  |  |  |
| --- | --- | --- |
| Prototype | int GetData(unsigned char ucDataType, const unsigned char \*pucTagList, unsigned int uiTagListLen,  unsigned int uiTLVListOutBufLen, unsigned char \*pucTLVListOut, unsigned int \*puiTLVListLenOut) | |
| Function | This API is use to set value for EMV/CLSS TAGs, transaction parameters, terminal parameters, and so on; | |
| **Parameters** | ucDataType [Input] | Data Type:  1: EMV Data.  2: Customize Data. See [Appendix 3 – Terminal Information List](#_Appendix_3_–) and [Appendix 4 – Application parameter list](#_Appendix_4_–). |
| \* pucTagList [Input] | List of requested tags. All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “\x9F\x26\x9F\x27\x4F” |
| uiTagListLen[Input] | The length of the tag list data. |
| uiTLVListOutBufLen[Input] | The length of the TLV list output buffer. |
| \* pucTLVListOut [Output] | List out TAGs which is success.  All the tags are consecutive without any separator.  e.g.  Request for tag 0x9F26 0x9F27 0x4F, then this field would be: “\x9F\x26\x9F\x27\x4F” |
| \*puiTLVListLenOut  [Output] | Length of successful TAGs list |
| **Return** |  |  |
| **Instruction** | Note: Tag 0x5A, 0x56, 0x9F6B, 0x57 are the sensitive tags, if get these tags, should encrypt it before send out. | |



## FileDownload

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int DownloadFile(const uchar \* pucRequest);** | |
| **Function** | this API is used to download parameter files, including EMV parameter file, EMV contactless parameter file, UI XML files, font, application, application parameter files from master device; | |
| **Parameters** | \* pucRequest [Input] | The request data, its format should be:  FileName  FileSize  FileOffset  TransferDataLen  FileData  For detail, see [File download](#_File_download) |
| **Return** |  |  |
| **Instruction** |  | |

## TMS Proxy



### GetTermSN

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int GetTermSN(unsigned char \*pucRspDataOut, int \*piLenOut)** | |
| **Function** | Get terminal SN; | |
| **Parameters** | \*pucRspDataOut [Onput] |  |
| \* piLenOut [Output] |  |
| **Return** |  |  |
| **Instruction** | For response data format, see [GetTermSN](#_GetTermSN) | |

### GetTermExtSN

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int GetTermExtSN (unsigned char \*pucRspDataOut, int \*piLenOut)** | |
| **Function** | Get terminal external SN; | |
| **Parameters** | \*pucRspDataOut [Onput] |  |
| \* piLenOut [Output] |  |
| **Return** |  |  |
| **Instruction** | For response data format, see [GetTermExtSN](#_GetTermExtSN) | |

### GetTermVerInfo

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int GetTermVerInfo (unsigned char \*pucRspDataOut, int \*piLenOut)** | |
| **Function** | Get version information of terminal; | |
| **Parameters** | \*pucRspDataOut [Onput] |  |
| \* piLenOut [Output] |  |
| **Return** |  |  |
| **Instruction** | For response data format, see [GetTermVerInfo](#_GetTermVerInfo) | |

### GetTerminalInfo

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int GetTerminalInfo (unsigned char \*pucRspDataOut, int \*piLenOut)** | |
| **Function** | Get information of terminal; | |
| **Parameters** | \*pucRspDataOut [Onput] |  |
| \* piLenOut [Output] |  |
| **Return** |  |  |
| **Instruction** | For response data format, see [GetTerminalInfo](#_GetTerminalInfo) | |

### RemoteDownload\_TaskInfo

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int RemoteDownload\_TaskInfo(const unsigned char \*pucReqData, unsigned char \*pucRspData, int \*piRspLen);** | |
| **Function** | Get information of terminal; | |
| **Parameters** | \*pucReqData[Input] | Request data |
| \* pucRspData[output] |  |
| \* piRspLen [Output] |  |
| **Return** |  |  |
| **Instruction** | For request and response data format, see [SetTaskList](#_SetTaskList) | |

### RemoteDownload\_SaveData

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int RemoteDownload\_SaveData (const unsigned char \*pucReqData, unsigned char \*pucRspData, int \*piRspLen);** | |
| **Function** | Get information of terminal; | |
| **Parameters** | \*pucReqData[Input] | Request data |
| \* pucRspData[output] |  |
| \* piRspLen [Output] |  |
| **Return** |  |  |
| **Instruction** | For request and response data format, see [SaveFileData](#_SaveFileData) | |

### RemoteDownload\_TaskAsk

|  |  |  |
| --- | --- | --- |
| **Prototype** | **int RemoteDownload\_TaskAsk(const unsigned char \*pucReqData, unsigned char \*pucRspData, int \*piRspLen, unsigned char \*pucSendRspFlag);** | |
| **Function** | Get information of terminal; | |
| **Parameters** | \*pucReqData[Input] | Request data |
| \* pucRspData[output] |  |
| \* piRspLen [Output] |  |
|  | \*pucSendRspFlag[output] | Send response flag: 0x00-send in communication module, 0x01-send in RemoteDownload\_TaskAsk function; |
| **Return** |  |  |
| **Instruction** | For request and response data format, see [GetTaskList](#_GetTaskList) | |



## File system

Same with PDK file system.

# Mandatory UI definition in POS terminal

There are some forms that need to be shown under some specific conditions.

Below are the mandatory UI definitions in POS terminal for EasyLink.



## Prompt for card

### Swipe/Insert/Tap Card

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This step shows a prompt to insert/swipe/tap card at the beginning of a new transaction |
| Pre-conditions | Magnetic/EMV Contact/EMV Contactless are all supported |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60s  Send timeout message to master device and end the transaction |
| Sample Prompt | English – SWIPE/INSERT/TAP CARD |
| Mandatory | YES |

Sample:





### Fallback Swipe

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This step shows a prompt to swipe a card when fallback is needed |
| Pre-conditions | * Specific error happens, such as EMV data error, etc. * Fallback Supported |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60s  Send timeout message to master device and end the transaction |
| Sample Prompt | English –SWIPE CARD  FALLBACK |
| Mandatory | YES |

Sample:



### Tap Card Again

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction/CompleteTransaction |
| Description | This step shows a prompt to tag card again in contactless transaction. |
| Pre-conditions | * The card was not held in the read field for the required time and the reader was not able to capture all data to complete the transaction. * if the transaction is need to check the CVM on mobile phone, and after display "See Phone" or same message * If there is issuer script data or issuer authentication data that need to be processed for online contactless transaction. |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60s  Send timeout message to master device and end the transaction |
| Sample Prompt | English –PLEASE TAP AGAIN |
| Mandatory | YES |

Sample:



### Remove Card

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction/CompleteTransaction |
| Description | This UI is used to prompt the cardholder to remove the card after transaction being terminated (error happened or transaction declined) or being completed (transaction approved).  For that the cardholder may forget to pull out the card after the transaction being ended, it's strongly recommended to beep a noise to alarm the cardholder to do that. |
| Pre-conditions | The Transaction has been finished |
| Page Name |  |
| Widget(s) |  |
| Timeout | None |
| Sample Prompt | English –PLS REMOVE CARD |
| Mandatory | YES |

Sample:



## Processing

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction/CompleteTransaction |
| Description | This UI is used to show that the kernel is reading EMV data or processing the transaction.  It's recommended that the terminal show a prompt that the cardholder shall not to remove card during the processing of the EMV transaction. Otherwise it would lead the transaction to be failed. |
| Pre-conditions | The Transaction has been finished |
| Page Name |  |
| Widget(s) |  |
| Timeout | None |
| Sample Prompt | English –PROCESSING…  PLS NOT REMOVE |
| Mandatory | YES |

Sample:



## See Phone

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This UI is used to prompt the cardholder to check the display on the screen of the mobile phone |
| Pre-conditions |  |
| Page Name |  |
| Widget(s) |  |
| Timeout | 5s |
| Sample Prompt | English – SEE PHONE |
| Mandatory | YES |

Sample:



## Application Selection

### Select EMV application

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This UI is used to display the application candidate list if there are more than one applications that are supported by both terminal and card.  It may also be possible to show this UI for cardholder to confirm the selected application. |
| Pre-conditions |  |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60s |
| Sample Prompt | English –  PLS SELECT:   1. APP 1 2. APP 2   … |
| Mandatory | YES |

Sample:





### Select EMV application again

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This UI is used to prompt the cardholder that the previous application selection is not effective, and cardholder shall try to select another application again.  After timeout of this UI, the previously not accepted application will be deleted from the candidate list. And if there is still application(s) in the candidate list, the UI [Select EMV application](#_Select_EMV_application) will be shown again. Otherwise, the transaction will be terminated. |
| Pre-conditions |  |
| Page Name |  |
| Widget(s) |  |
| Timeout | 3s |
| Sample Prompt | English –  NOT ACCEPT!  PLS TRY AGAIN! |
| Mandatory | YES |

Sample:



## PIN Entry

### Enter PIN

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This UI is used to prompt the cardholder to enter PIN. |
| Pre-conditions | When performing CVM, offline (Plaintext/Cipher) PIN or online PIN is needed |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60S |
| Sample Prompt | English –  PLS ENTER PIN:  \*\*\*\*\*\* |
| Mandatory | YES |

Sample:





### Enter Offline PIN Again

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This UI is used to prompt the cardholder that the previously entered PIN was wrong. And the cardholder need to try inputting PIN again |
| Pre-conditions | Offline PIN is supported |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60S |
| Sample Prompt | English –  PIN ERROR!  PLS TRY AGAIN! |
| Mandatory | YES |

Sample:



### Last Chance to Enter PIN

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This step prompts the cardholder that there is only one chance left to enter offline PIN, otherwise the card will be blocked. |
| Pre-conditions | Offline PIN is supported |
| Page Name |  |
| Widget(s) |  |
| Timeout | 60S |
| Sample Prompt | English –  LAST ENTER PIN  PLS ENTER PIN |
| Mandatory | YES |

Sample:



### PIN Verify OK

|  |  |
| --- | --- |
| Transaction Step | During StartTransaction |
| Description | This step prompts the cardholder that offline PIN has been verified correctly |
| Pre-conditions | When performing CVM, offline PIN was verified successfully |
| Page Name |  |
| Widget(s) |  |
| Timeout | none |
| Sample Prompt | English –PIN VERIFY OK |
| Mandatory | NO |

Sample:



## Amount Confirm

|  |  |
| --- | --- |
| 1. Transaction Step | During GetPinBlock/StartTransaciton |
| Description | This UI is used to prompt the cardholder to confirm the transaction amount |
| Pre-conditions |  |
| Page Name |  |
| Widget(s) |  |
| Timeout | 5s |
| Sample Prompt | English –  Total amount:  10.00 KD  CORRECT? Y/N |
| Mandatory | YES |

Sample:



# Terminal configuration

## XML file data format

Hex: HEX characters (0~9, A~E).

N: Numeric characters (0~9).

B: 0 or 1.

Char: Any available ASCII characters.

Hex 10: Data length fixed with 10 characters. For Hex format data, It means 5 bytes of Hex data(refer to the example below).

Hex…496: Data length maximumly being 496 characters (258 bytes of Hex data).

Example:

RID (Hex 10):<RID>A000000005<RID> -- The RID is “\xA0\x00\x00\x00\x05”

ExpDate(N 6):<ExpDate>140710<ExpDate> -- The expiration date is “140710”

AppName (Char…16):<AppName>MCHIP<AppName> -- The maximum length of application name is 16 ASCII characters

## EMV parameter XML file configuration

### EMV AID configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | PartialAIDSelection | M | B   1 | Partial AID Selection Flag  The value range is shown as below:  0: Partial Match  1: Full Match |
| 2 | ApplicationID | M | Hex…34 | Application Identifier |
| 3 | IfUseLocalAIDName | M | B 1 | The value is as below:  0: Use Application   Name From Card  1: Use Application   Local Name |
| 4 | LocalAIDName | M | Char…16 | Local Application Name |
| 5 | TerminalAIDVersion | M | Hex 4 | Application Version |
| 6 | TACDenial | M | Hex 10 | TAC Denial |
| 7 | TACOnline | M | Hex 10 | TAC Online |
| 8 | TACDefault | M | Hex 10 | TAC Default |
| 9 | IfUseTerminalDefaultDDOL | M | N 1 | The value range is shown as below:  0: Don’t Use Terminal Default DDOL  1: Use Terminal   Default DDOL |
| 10 | TerminalDefaultDDOL | M | Hex…256 | Terminal   Default DDOL |
| 11 | FloorLimit | M | N…12 | Terminal Floor   Limit |
| 12 | Threshold | M | N…12 | Threshold |
| 13 | TargetPercentage | M | N…2 | Target Percentage |
| 14 | MaxTargetPercentage | M | N…2 | Maximum Target Percentage |
| 15 | TerminalDefaultTDOL | M | Hex…512 |  |
| 16 | TerminalDefaultDDOL | M | Hex...512 |  |
| 17 | TerminalRiskManagementData | M | Hex...10 | Provided by issuer.  No need to set this until issuer requested. |

Note: the AID parameter shall be configured into XML file.

### EMV CAPK

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | RID | M | Hex   10 | Application Service Provider ID |
| 2 | KeyID | M | Hex 2 | Key Index |
| 3 | HashArithmeticIndex | M | N 1 | HASH Flag |
| 4 | RSAArithmeticIndex | M | N 1 | RSA Flag |
| 5 | ModuleLength | M | N…4 | Module Length |
| 6 | Module | M | Hex…496 | Module |
| 7 | ExponentLength | M | N1 | Exponent Length |
| 8 | Exponent | M | Hex…6 | Exponent |
| 9 | ExpireDate | M | N 6 | Expiration Date (YYMMDD) |
| 10 | CheckSum | M | Hex 40 | Key Check Sum |

Note: the CAPK parameter shall be configured into XML file.

### EMV Revoked CAPK

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | RID | M | Hex   10 | Application Service Provider ID |
| 2 | KeyID | M | Hex 2 | Key Index |
| 3 | CertificateSN | M | Hex…6 | Issuer Certificate Serial No. |

Note: the revoked CAPK parameter shall be configured into XML file.

### ICS (Implementation Conformance Statement) Configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | Type | M | ans...128 | ICS configuration type.  Currently there are 12 different configurations available for PX terminals. All the different ICS configurations can be pre-configured in the XML parameter file. This field "Type" defines an specific type of ICS configuration. And ECR-POS can designate any one of them to be applied for the current transaction according to different scenarios. |
| 2 | TerminalType | M | Hex 2 | Terminal type. Below is the supported terminal type according to PX EMV L2 certificates:  22: attended, online with offline capability terminal  25: unattended, online with offline capability terminal |
| 3 | CardDataInputCapability | M | Hex 2 | Card Data Input Capability (2 characters, i.e. 1 byte HEX data). Each bit represents a flag of a specific capability describing the capability of the terminal to capture card data: bit set to 1 means that the capability is supported while 0 means not supported. Below is the representation of each bit :  (bit 8 indicates the most significant bit - i.e. the leftmost bit)  b1~b5: Reserved  b6: IC with contacts  b7: Magnetic stripe  b8: Manual key entry |
| 4 | CVMCapability | M | Hex 2 | CVM Capability (2 characters, i.e. 1 byte HEX data). Each bit represents a flag of a specific capability describing the capability of the terminal to perform CVM (Cardholder Verification Method): bit set to 1 means that the capability is supported while 0 means not supported. Below is the representation of each bit:  (bit 8 indicates the most significant bit - i.e. the leftmost bit)  b1~b3: Reserved  b4: No CVM  b5: Enciphered PIN for offline ICC verification  b6: Signature (paper)  b7: Enciphered PIN for online verification  b8: Plaintext PIN for offline ICC verification |
| 5 | SecurityCapability | M | Hex 2 | Security Capability (2 characters, i.e. 1 byte HEX data). Each bit represents a flag of a specific capability describing the security capability of the terminal: bit set to 1 means that the capability is supported while 0 means not supported. Below is the representation of each bit:  (bit 8 indicates the most significant bit - i.e. the leftmost bit)  b1~b3: Reserved  b4: CDA (Combined DDA/Application Cryptogram Generation)  b5: Reserved  b6: Card Capture (Always be 0 for POS terminal)  b7: DDA (Dynamic Data Authentication)  b8: SDA (Static Data Authentication) |
| 6 | AdditionalTerminalCapabilities | M | Hex 10 | Additional Terminal Capabilities (10 characters, i.e. 5 byte HEX data). Each bit (totally 40 bits) represents a flag of a specific additional terminal capability: bit set to 1 means that the capability is supported while 0 means not supported. Below is the representation of each bit:  (byte 1 indicates the leftmost byte)  (bit 8 indicates the most significant bit - i.e. the leftmost bit)  Byte 1 - Transaction Type Capability  b1: Administrative  b2: Payment  b3: Transfer  b4: Inquiry  b5: Cashback  b6: Services  b7: Goods  b8: Cash  Byte 2 - Transaction Type Capability  b1 ~ bit7: Reserved  bit 8: Cash Deposit  Byte 3 - Terminal Data Input Capability  b1 ~ b4: Reserved  b5: Function keys  b6: Command keys  b7:Alphabetic and special characters keys  b8: Numeric keys  Byte 4 - Terminal Data Output Capability  b1: Code table 9  b2: Code table 10  b3~b4: Reserved  b5: Display, cardholder  b6: Display, attendant  b7: Print, cardholder  b8: Print, attendant  Byte 5 - Terminal Data Output Capability  b1: Code table 1  b2: Code table 2  b3: Code table 3  b4: Code table 4  b5: Code table 5  b6: Code table 6  b7: Code table 7  b8: Code table 8  (Note: The code table number refers to the corresponding part of ISO/IEC 8859.) |
| 7 | GetDataForPINTryCounter | M | N 1 | Indicates if supported get the PIN retry counter before let cardholder enter offline PIN.  0 - NOT supported  1 - supported |
| 8 | BypassPINEntry | M | N 1 | Indicates if bypass PIN entry is allowed  0 - NOT supported  1 - supported |
| 9 | SubsequentBypassPINEntry | M | N 1 | Indicates if subsequent bypass PIN entry is allowed  0 - NOT supported  1 - supported |
| 10 | ExceptionFileSupported | M | N 1 | indicates if check exception file is supported  0 - NOT supported  1 - supported |
| 11 | ForcedOnlineCapability | M | N 1 | indicates if merchant force transaction online is allowed  0 - NOT supported  1 - supported |
| 12 | IssuerReferralsSupported | M | N 1 | Indicates if referral supported  0 - NOT supported  1 - supported |
| 13 | ConfigurationCheckSum | M | Hex 8 | Configuration checksum.  This checksum is used to make sure that the configuration that is being used is configured according to the EMV L2 certificates. If the checksum is wrong, the transaction will be terminated. |

### Terminal common EMV configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | MerchantID | M | ans...15 | Merchant Identification |
| 2 | TerminalID | M | ans 8 | Terminal Identification |
| 3 | TerminalCountryCode | M | Hex   4 | Terminal Country Code |
| 4 | TerminalCurrencyCode | M | Hex 4 | Terminal Currency Code |
| 5 | ReferenceCurrencyCode | M | Hex 4 | Reference Currency Code |
| 6 | TerminalCurrencyExponent | M | Hex 2 | Terminal Currency Exponent |
| 7 | ReferenceCurrencyExponent | M | Hex 2 | Reference Currency Exponent |
| 8 | ConversionRatio | M | n...6 | the conversion quotients between transaction currency and reference currency (default : 1000)  (the exchange rate of transaction currency to reference currency \*1000) |
| 1 | MerchantID | M | ans...15 | Merchant Identification |
| 2 | TerminalID | M | ans 8 | Terminal Identification |
| 3 | TerminalCountryCode | M | Hex   4 | Terminal Country Code |
| 4 | TerminalCurrencyCode | M | Hex 4 | Terminal Currency Code |
| 5 | ReferenceCurrencyCode | M | Hex 4 | Reference Currency Code |
| 6 | TerminalCurrencyExponent | M | Hex 2 | Terminal Currency Exponent |
| 7 | ReferenceCurrencyExponent | M | Hex 2 | Reference Currency Exponent |
| 8 | ConversionRatio | M | n...6 | the conversion quotients between transaction currency and reference currency (default : 1000)  (the exchange rate of transaction currency to reference currency \*1000) |

Note: the EMV common configuration parameter shall be configured into XML file.

## Contactless EMV parameter XML file configuration

All the EMV contactless parameters shall be configured into XML file.

### Paywave parameter configuration

#### Paywave configuration

Reserved.

#### Paywave AID configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | PartialAIDSelection | M | N   1 | Select Flag  The value is as below:  0: Partial Match  1: Full Match |
| 2 | ApplicationID | M | Hex…34 | Application Identifier |
| 3 | IfUseLocalAIDName | M | N 1 | The value is as below:  0: Use Application   Name From Card  1: Use Application   Local Name |
| 4 | LocalAIDName | M | Char…16 | Local Application Name |
| 5 | TerminalAIDVersion | M | Hex 4 | Application Version |
| 6 | CryptogramVersion17Supported | M | N…1 | MSD   CVN17 support flag,  0-      not support   1. 1-      support |
| 7 | ZeroAmountNoAllowed | M | N…1 | Amount, Authorized of Zero Check]  flag,  0-        [Amount,   Authorized of Zero Check] activated, online required  1-        [Amount,   Authorized of Zero Check] activated, amount zero not allowed  2-    [Amount, Authorized of Zero Check] deactivated |
| 8 | StatusCheckSupported | M | N…1 | status   check support flag,   0-not support ，1-   support |
| 9 | ReaderTTQ | M | Hex…8 | Indicates reader   capabilities, requirements, and preferences to the card.  Byte   1  bit 8: 1 = MSD supported  bit 7: RFU (0)  bit 6: 1 = qVSDC   supported  bit 5: 1 = EMV contact   chip supported  bit 4: 1 = Offline-only   reader  bit 3: 1 = Online PIN   supported  bit 2: 1 = Signature   supported  bit 1: 1 = Offline Data   Authentication (ODA) for Online Authorizations supported.  Note: Readers compliant to this specification   set TTQ byte 1 bit 1 to 0b  Byte 2  bit 8: 1 = Online   cryptogram required  bit 7: 1 = CVM   required  bit 6: 1 = (Contact   Chip) Offline PIN supported  bits 5-1: RFU   (00000)  Byte 3  bit 8: 1 = Issuer   Update Processing supported  bit 7: 1 = Mobile   functionality supported (Consumer Device CVM)  bits 6-1: RFU   (000000)  Byte 4  RFU ('00') |
| 10 | Inter\_WareFloorlimitByTransactionType | M | Node | Sub Node,   refer to “2.1.2.2.1 Sub Node  Inter\_WareFloorlimitByTransactionType   description” |

##### Sub Node “Inter\_WareFlmtByTransType” description

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | ContactlessCVMLimit | M | N…12 | Contactless CVM Floor   Limit |
| 2 | ContactlessTransactionLimit | M | N…12 | Contactless Transaction Floor Limit |
| 3 | ContactlessFloorLimit | M | N…12 | Contactless Floor Limit |
| 4 | TerminalFloorLimit | M | N…12 | Terminal Floor Limit |
| 5 | TerminalFloorLimitSupported | M | N…1 | Terminal   Offline limit check flag  0-Deactivated, 1-Active and exist, 2-Active but not exist |
| 6 | ContactlessTransactionLimitSupported | M | N…1 | Card reader contactless transaction limit check flag,   0-Deactivated, 1-Active and exist, 2-Active but not exist |
| 7 | CVMLimitSupported | M | N…1 | Card reader CVM limit check flag, 0-Deactivated, 1-Active and   exist, 2-Active but not exist |
| 8 | ContactlessFloorLimitSupported | M | N…1 | Card reader contactless Offline limit check flag,   0-Deactivated, 1-Active and exist, 2-Active but not exist |
| 9 | TransactionType | M | N2 | Transaction type   same as Tag“0x9C” |

#### Program ID configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | ProgramId | M | N…17 | Program ID |
| 2 | ContactlessCVMLimit | M | N…12 | Contactless CVM Floor Limit |
| 3 | ContactlessTransactionLimit | M | N…12 | Contactless   Transaction Floor Limit |
| 4 | ContactlessFloorLimit | M | N…12 | Contactless   Floor Limit |
| 5 | TerminalFloorLimit | M | N…12 | Terminal Floor   Limit |
| 6 | TerminalFloorLimitSupported | M | N…1 | Terminal Offline limit check flag  0-Deactivated, 1-Active and   exist, 2-Active but not exist |
| 7 | ContactlessTransactionLimitSupported | M | N…1 | Card reader contactless   transaction limit check flag, 0-Deactivated, 1-Active and exist, 2-Active but   not exist |
| 8 | CVMLimitSupported | M | N…1 | Card reader CVM limit check   flag, 0-Deactivated, 1-Active and exist, 2-Active but not exist |
| 9 | ContactlessFloorLimitSupported | M | N…1 | Card reader contactless   Offline limit check flag, 0-Deactivated, 1-Active and exist, 2-Active but not   exist |
| 10 | CryptogramVersion17Supported | M | N…1 | MSD CVN17 support flag,  0 - not support  1 - support |
| 11 | ZeroAmountNoAllowed | M | N…1 | Amount, Authorized of Zero   Check]  flag,  0 - [Amount, Authorized of   Zero Check] activated, online required  1 - [Amount, Authorized of   Zero Check] activated, amount zero not allowed  2 - [Amount, Authorized of   Zero Check] deactivated |
| 12 | StatusCheckSupported | M | N…1 | status check support flag,   0-not support ，1- support |
| 13 | ReaderTTQ | M | Hex…8 | Indicates reader capabilities, requirements, and   preferences to the card.  Byte 1  bit 8: 1 = MSD supported  bit 7: RFU (0)  bit 6: 1 = qVSDC supported  bit 5: 1 = EMV contact chip supported  bit 4: 1 = Offline-only reader  bit 3: 1 = Online PIN supported  bit 2: 1 = Signature supported  bit 1: 1 = Offline Data Authentication (ODA) for Online   Authorizations supported.  Note: Readers compliant to this   specification set TTQ byte 1 bit 1 to 0b  Byte 2  bit 8: 1 = Online cryptogram required  bit 7: 1 = CVM required  bit 6: 1 = (Contact Chip) Offline PIN supported  bits 5-1: RFU (00000)  Byte 3  bit 8: 1 = Issuer Update Processing supported  bit 7: 1 = Mobile functionality supported (Consumer   Device CVM)  bits 6-1: RFU (000000)  Byte 4  RFU ('00') |

### Paypass parameter configuration

#### Paypass configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | KernelConfiguration | M | Hex…2 | DF811B             Indicates the Kernel configuration   options    b8:Only EMV mode   transactions supported  b7: Only mag-stripe   mode transactions supported  b6:On device   cardholder verification supported  b5-1 RFU  decide by AID |
| 2 | TornLeftTime | M | N…4 | DF811C             Maximum time, in seconds, that a   record can remain in the Torn Transaction Log.  (only for chip card) |
| 3 | MaximumTornNumber | M | Hex…2 | DF811D Max   Number of Torn Transaction Log Records  (only for chip card) |
| 4 | MagneticCVM | M | Hex…2 | DF811E Mag-stripe CVM Capability-CVM   required(PayPass)  b8-5  0000:   NO CVM  0001:   OBTAIN SIGNATURE  0010:   ONLINE PIN  1111:   N/A  b4-1  NFU |
| 5 | MageticNoCVM | M | Hex…2 | DF812C Mag-stripe CVM Capability-No CVM   required  b8-5  0000:   NO CVM  0001:   OBTAIN SIGNATURE  0010:   ONLINE PIN  1111:   N/A  b4-1  NFU |
| 6 | MobileSupport | M | Hex…2 | Tag 9F7E Mobile Support Indicator  b8-3： RFU  b2 :Offline   PIN Required  b1：Mobile supported |
| 7 | CardDataInput | M | Hex…2 | Tag DF8117  Card Data Input Capability |
| 8 | CVMCapability\_CVMRequired | M | Hex…2 | Tag DF8118  CVM Capability - CVM Required |
| 9 | CVMCapability\_NoCVMRequired | M | Hex…2 | Tag DF8119  CVM Capability - No CVM Required  (0x08) |
| 10 | TerminalType | M | Hex…2 | Tag  9F35 Terminal Type |
| 11 | AccoutType | M | Hex2 | Tag 5F57   Account type |
| 12 | AdditionalTerminalCapability | M | Hex10 | Tag   9F40 Additional Terminal Capbilities |
| 13 | KernelID | M | Hex 1 | Tag   DF810C Kernel ID |
| 14 | SecurityCapability | M | Hex 1 | Tag   DF811F security capability  b1~b3: reserved  b4: CDA  b5: reserved  b6: Card Capture (Always be 0)  b7: DDA  b8: SDA |

#### Paypass AID configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | PartialAIDSelection | M | N 1 | Select Flag  The value is as below:  0: Partial Match  1: Full Match |
| 2 | ApplicationID | M | Hex…34 | Application Identifier |
| 3 | IfUseLocalAIDName | M | N 1 | The value is as below:  0: Use Application   Name From Card  1: Use Application   Local Name |
| 4 | LocalAIDName | M | Char…16 | Local Application Name |
| 5 | TerminalAIDVersion | M | Hex 4 | Tag 9F09   :  Application Version |
| 6 | MagneticApplicationVersionNumber | M | Hex 4 | 9F6D Mag-stripe Application   Version Number (Reader) |
| 7 | TACDenial | M | Hex 10 | TAG DF8121 : TAC   Denial  (only for chip card) |
| 8 | TACOnline | M | Hex 10 | TAG DF8122:TAC   Online  (only for chip card) |
| 9 | TACDefault | M | Hex 10 | TAG DF8120: TAC   Default  (only for chip card) |
| 10 | TerminalRisk | M | Hex 16 | (Tag 9F1D)  B7:Plaintext PIN for ICC verification (Contactless)  B6:Enciphered PIN for online verification (Contactless)  B5:Signature (paper) (Contactless)  B4:Enciphered PIN for offline verification   (Contactless)  b3:No CVM required (Contactless)  b2:On device cardholder verification (Contactless)  (only for chip card) |
| 11 | ContactlessCVMLimit | M | N…12 | Tag DF8126 Contactless   CVM Floor Limit |
| 12 | ContactlessTransactionLimit\_NoOnDevice | M | N…12 | Tag DF8124 Indicates the transaction amount above which the   transaction is not allowed, when on-device cardholder verification is not   supported. |
| 13 | ContactlessTransactionLimit\_OnDevice | M | N…12 | Tag DF8125 Indicates the transaction amount above which the   transaction is not allowed, when on-device cardholder verification is   supported. |
| 14 | ContactlessFloorLimit | M | N…12 | Tag DF8123 Contactless Floor Limit |

#### Other contactless EMV parameter configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | CountryCode | M | Hex 4 | Terminal Country Code |
| 2 | CurrencyCode | M | Hex 4 | Terminal Currency Code |
| 3 | RefCurcyCode | M | Hex 4 | Reference Currency Code |
| 4 | CurrencyExp | M | Hex 2 | Terminal Currency Exponent |
| 5 | RefCurrencyExp | M | Hex 2 | Reference Currency Exponent |
| 6 | MerchantCategoryCode | M | Hex 4 | Classifies the type of business being done by the   merchant, represented in accordance with [ISO 8583:1993] for Card Acceptor   Business Code. |
| 7 | MerchantId | M | Asc…15 | When concatenated with the Acquirer Identifier,   uniquely identifies a given merchant. |
| 8 | TerminalID | M | N 8 | Designates the unique location of the Terminal. |
| 9 | MerchantName | M | ASC…128 | Indicates the name of the merchant. |
| 10 | MerchantLocalAddress | M | ASC…128 | Indicates the location of the merchant. |
| 11 | ReferenceCurrenceConverRate | M | N6 | The conversion quotients between transaction currency   and reference currency. (the exchange rate of transaction currency to   reference currency \*1000)  default : 1000 |

### ExpressPay parameter configuration

#### ExpressPay configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | TerminalSupportOptimizationModeTransaction | M | N1 | The terminal whether to support the optimization mode transaction  0: Not supported  1: Supported |
| 2 | UnpredictableNumberRange | M | N4 | The Range of Magnetic Stripe unpredictable Number(00-99) |
| 3 | TerminalTransactionCapability | M | Hex8 | 9F6E TerminalTransactionCapability  Byte 1:  Bit8: 1=AEIPS contact mode supported  Bit7: 1=Expresspay Magstripe Mode supported  Bit6: 1 = Expresspay EMV full online mode supported  Bit5: 1=Expresspay EMV partial online mode supported  Bit4: 1=Expresspay Mobile Supported  b3-b1: RFU  Byte 2:  Bit8: 1 = Mobile CVM supported  Bit7: 1 = Online PIN supported  Bit6: 1 = Signature  Bit5: 1 = Plaintext Offline PIN  B4-b1: RFU  Byte 3:  Bit8: 1 = Terminal is offline only  Bit7: 1 = CVM Required  B6-b1: RFU  Byte 4: RFU |
| 4 | DelayAuthorizationSupport | M | N1 | 1: support Delayed Authorization;  0: not support Delayed Authorization |

#### ExpressPay AID configuration

| **No.** | **Field Name** | **Required** | **Attribute** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | PartialAIDSelection | M | N 1 | Select Flag  The value is as below:  0: Partial Match  1: Full Match |
| 2 | ApplicationID | M | Hex…34 | Application Identifier |
| 3 | IfUseLocalAIDName | M | N 1 | The value is as below:  0: Use Application   Name From Card  1: Use Application   Local Name |
| 4 | LocalAIDName | M | Char…16 | Local Application Name |
| 5 | TerminalAIDVersion | M | Hex 4 | Tag 9F09   :  Application Version |
| 6 | TACDenial | M | Hex 10 | TAG DF8121 : TAC   Denial  (only for chip card) |
| 7 | TACOnline | M | Hex 10 | TAG DF8122:TAC   Online  (only for chip card) |
| 8 | TACDefault | M | Hex 10 | TAG DF8120: TAC   Default  (only for chip card) |
| 9 | ExpresspayTerminalCapabilities | M | Hex2 | Tag 9F6D:  Expresspay Terminal Capability  Bit(8-7)  00: Expresspay 1.0  01: Expresspay 2.0 MagStrip Only  10: Expresspay 2.0  EMV and MagStrip  11: Expresspay Mobile  EMV Supported |
| 10 | DDOL | M | Hex…256 | Terminal Default Dynamic Data. Authentication Data Object List |
| 11 | TDOL | M | Hex…256 | Terminal Default Data Object List |
| 12 | ContactlessCVMLimit | M | N…12 | Contactless CVM Limit |
| 13 | ContacltessTransactionLimit | M | N…12 | Indicates the tranasaction amount above which the transaction is not allowed |
| 14 | ContactlessFloorLimit | M | N…12 | Contactless Floor Limit |

## UI XML file configuration

The UI layout shall be configured into XML file.

### TextBox

|  |  |
| --- | --- |
| Property name | Description |
| Name | Widget name |
| Row | Row number |
| Col | Column number |
| TextAlign | Text align:  ALIGN\_LEFT,  ALIGN\_CENTER,  ALIGN\_RIGHT |
| Mode | Display property:  0x00-normal  0x01-reserse |
| FontSize | Font size |
| Type | Textbox type:  0-text,  1-picture |
| Text | Text content |
| Picture | Picture content |
| Key accept | Key accept:  BUTTON\_NONE,  BUTTON\_OK,  BUTTON CANCEL,  BUTTON\_OK\_AND\_CANCEL, |

### InputBox

|  |  |
| --- | --- |
| Property name | Description |
| Name | Widget name |
| Row | Row number |
| Col | Column number |
| TextAlign | Text align:  ALIGN\_LEFT,  ALIGN\_CENTER,  ALIGN\_RIGHT |
| Mode | Display property:  0-normal  1-mask |
| FontSize | Font size |
| Text | Content |

### Menu

|  |  |
| --- | --- |
| Property name | Description |
| Name | Widget name |
| Row | Row number |
| Col | Column number |
| FontSize | Font size |
| TextAlign | Text align:  ALIGN\_LEFT,  ALIGN\_CENTER,  ALIGN\_RIGHT |

# Appendix

## Appendix 1 – Return code

### Base return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_RET\_OK | 0 | success |
| EL\_COMM\_RET\_BASE | 1000 | Base return code for COMM module |
| EL\_UI\_RET\_BASE | 2000 | Base return code for UI module |
| EL\_SECURITY\_RET\_BASE | 3000 | Base return code for Security module |
| EL\_TRANS\_RET\_BASE | 4000 | Base return code for transction module |
| EL\_PARAM\_RET\_BASE | 5000 | Base return code for parameter management module |
| EL\_COMMON\_RET\_BASE | 6000 | Base return code for common module |
| EL\_FILEDOWNLOAD\_RET\_BASE | 7000 | Base return code for FileDownload module |

### COMM return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_COMM\_RET\_BASE | 1000 |  |
| EL\_COMM\_RET\_CONNECTED | (EL\_COMM\_RET\_BASE + 1) | Already connected |
| EL\_COMM\_RET\_DISCONNECT\_FAIL | (EL\_COMM\_RET\_BASE + 2) | Disconnect fail |
| EL\_COMM\_RET\_NOT\_CONNECTED | (EL\_COMM\_RET\_BASE + 3) | Not connected |
| EL\_COMM\_RET\_OPEN\_COMPATIBLE\_MODE\_FAIL | (EL\_COMM\_RET\_BASE + 4) | Open compatible communication mode failed |

### UI return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_UI\_ERT\_BASE | 2000 |  |
| EL\_UI\_RET\_INVALID\_PARAM | (EL\_UI\_ERT\_BASE + 1) | Invalid parameter |
| EL\_UI\_RET\_MEM\_RUN\_OUT | (EL\_UI\_RET\_BASE + 2) | Memory run out |
| EL\_UI\_RET\_TIME\_OUT | (EL\_UI\_ERT\_BASE + 3) | Time out |
| EL\_UI\_RET\_INVALID\_CONFIG | (EL\_UI\_ERT\_BASE + 4) | Invalid config |
| EL\_UI\_RET\_INVALID\_PAGE | (EL\_UI\_ERT\_BASE + 5) | Invalid page |

### Security return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_SECURITY\_RET\_BASE | 3000 |  |
| EL\_SECURITY\_RET\_NO\_KEY | (EL\_SECURITY\_RET\_BASE + 1) | Key does not exist |
| EL\_SECURITY\_RET\_INVALID\_PARAM | (EL\_SECURITY\_RET\_BASE + 2) | Parameter error or invalid. |
| EL\_SECURITY\_RET\_ENCRYPT\_DATA\_ERR | (EL\_SECURITY\_RET\_BASE +3) | Encrypt data error |
| EL\_SECURITY\_RET\_GET\_PIN\_BLOCK\_ERR | (EL\_SECURITY\_RET\_BASE + 4) | Get pin block error |
| EL\_SECURITY\_RET\_NO\_PIN\_INPUT | (EL\_SECURITY\_RET\_BASE + 5) | No input pin |
| EL\_SECURITY\_RET\_INPUT\_CANCEL | (EL\_SECURITY\_RET\_BASE + 6) | User cancel |
| EL\_SECURITY\_RET\_INPUT\_TIMEOUT | (EL\_SECURITY\_RET\_BASE + 7) | Input timeout |

### Transaction flow return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_TRANS\_RET\_BASE | 4500 |  |
| EL\_TRANS\_RET\_FALLBACK | (EL\_TRANS\_RET\_BASE + 1) | Request fallback swipe |
| EL\_TRANS\_RET\_ERR\_TRAN\_FAIL | (EL\_TRANS\_RET\_BASE + 2) | Transaction failed, including some general error, get the specific error code by calling GetData to get GetLastError. |
| EL\_TRANS\_RET\_USERCANCEL | (EL\_TRANS\_RET\_BASE + 3) | User canceled. |
| EL\_TRANS\_RET\_UNSUPPORT\_CARD | (EL\_TRANS\_RET\_BASE + 4) | The card is not supported. |
| EL\_TRANS\_RET\_CARD\_DATA\_ERR | (EL\_TRANS\_RET\_BASE + 5) | Card data error. |
| EL\_TRANS\_RET\_NO\_APP | (EL\_TRANS\_RET\_BASE + 6) | No corresponding application. |
| EL\_TRANS\_RET\_DECLINED | (EL\_TRANS\_RET\_BASE + 7) | Transaction declined. |
|  |  |  |
| EL\_TRANS\_RET\_ |  |  |

### Parameter management return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_PARAM\_RET\_BASE | 5000 |  |
| EL\_PARAM\_RET\_ERR\_DATA | (EL\_PARAM\_RET\_BASE + 1) | Input data error. |
| EL\_PARAM\_RET\_INVALID\_PARAM | (EL\_PARAM\_RET\_BASE + 2) | Invalid parameter. |
| EL\_PARAM\_RET\_PARTIAL\_FAILED | (EL\_PARAM\_RET\_BASE + 3) | Partial operation failed. |
| EL\_PARAM\_RET\_ALL\_FAILED | (EL\_PARAM\_RET\_BASE + 4) | All operation failed. |
| EL\_PARAM\_RET\_BUFFER\_TOO\_SMALL | (EL\_PARAM\_RET\_BASE + 5) | The output buffer size is not enough. |
| EL\_PARAM\_RET\_API\_ORDER\_ERR | (EL\_PARAM\_RET\_BASE + 6) | Must call this function after StartTransaction step. |

### TMS Proxy return code

| Return code | Value | Description |
| --- | --- | --- |
| MPOS\_STATUS\_OK | 0 | Success |
| MPOS\_STATUS\_CMD\_NOTSPT | -0xffff | Command not supported |
| MPOS\_STATUS\_MONITOR\_SAVE\_ERR | 5 | Save monitor error |
| MPOS\_STATUS\_FONT\_SAVE\_ERR | 6 | Save font error |
| MPOS\_STATUS\_APP\_SAVE\_ERR | 7 | Save application error |
| MPOS\_STATUS\_PARA\_SAVE\_ERR | 8 | Save parameter file error |
| MPOS\_STATUS\_NOTASKLIST | 1 | No task list |

### FileDownload return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_FILEDOWNLOAD\_RET\_BASE | 7000 |  |
| EL\_FILEDOWNLOAD\_RET\_INVALID\_PARAM | (EL\_FILEDOWNLOAD\_RET\_BASE + 1) | Invalid parameter |
| EL\_FILEDOWNLOAD\_RET\_PARAM\_FILE\_FAIL | (EL\_FILEDOWNLOAD\_RET\_BASE + 2) | Download parameter file failed |
| EL\_FILEDOWNLOAD\_RET\_FIRMWARE\_FAIL | (EL\_FILEDOWNLOAD\_RET\_BASE + 3) | Download firmware/app/font failed |
| EL\_FILEDOWNLOAD\_RET\_FILE\_OVERSIZE | (EL\_FILEDOWNLOAD\_RET\_BASE + 4) | File oversize:  Parameter file size: from 0K to 100K;  Firmware/app file size: from 0K to 400K;  Font file size: from 0K to 2048K; |
|  |  |  |

### Common return code

| Return code | Value | Description |
| --- | --- | --- |
| EL\_COMMON\_RET\_BASE | 6000 |  |
| EL\_COMMON\_RET\_INVALID\_PARAM | (EL\_COMMON\_RET\_BASE + 1) | Invalid parameter |
| EL\_COMMON\_RET\_FILE\_SIZE\_OVER\_LIMIT | (EL\_COMMON\_RET\_BASE + 2) | File size over limit |
| EL\_COMMON\_RET\_FILE\_TYPE\_NOT\_SPT | (EL\_COMMON\_RET\_BASE + 3) | File type not supported |
| EL\_COMMON\_RET\_FILE\_SAVE\_ERROR | (EL\_COMMON\_RET\_BASE + 4) | Save file data error |
|  |  |  |

## Appendix 2 – Table file block distribution

The default size of table file block is 1024 bytes.

| Block Index | Description |
| --- | --- |
| 0 ~ 49 | For Application parameter use |
|  |  |
| 50 ~ 199 | For UI configuration use |
|  |  |
| 200 ~ 299 | For CAPK use |
| 300 ~ 399 | For Revoke CAPK use |
|  |  |
| 400 ~ 499 | For EMV AID use |
| 500 ~ 509 | For EMV merchant parameter( EMV\_PARAMETER) use |
| 510 | For EMV\_EXTMPARAM use |
| 511 | For EMV\_MCKPARAM use |
|  |  |
| 550 ~ 649 | For PayWave AID use |
| 650 ~ 749 | For PayWave program ID configuration (only for chip chad) use |
|  |  |
| 750 ~ 849 | For PayPass AID use |
| 850 ~ 899 | For PayPass configuration |
| 900 ~ 949 | For Paypass Torn Log |
|  |  |
| 1000 ~ 1049 | For TLV data saving. |
|  |  |
| 3000 ~ 3299 | Temporary buffer when download EMV parameter file |
| 3300 ~ 3599 | Temporary buffer when download contactless EMV parameter file |
| 3600 ~ 3899 | Temporary buffer when download UI configuration file |

## Appendix 3 – Terminal Information List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAG** | **Element** | **Attribute** | **Read Write** | **Description** |
| 0101 | TermSN | ans…32 | R | Terminal SN.  The returned serial number is a string which ending with ‘\0‘.  The maximum length is 32 bytes. If SerialNo[0] = ‘\0‘, then it means there is no serial number.  Only 8 bytes are used currently, all are digits. |
| 0102 | TermModelCode | ans…6 | R | Indicates terminal model code, such as: “S80”, “S900” etc. |
| 0103 | TermPrinterInfo | a 1 | R | Indicates the printer type,  ‘S’ - Stylus printer.  ‘T‘- Thermal printer. |
| 0104 | TermModemExist | an 1 | R | Return the Modem module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0105 | TermUSBExist | an 1 | R | Return the USB module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0106 | TermLANExist | an 1 | R | Return the LAN module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0107 | TermGPRSExist | an 1 | R | Return the GPRS module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0108 | TermCDMAExist | an 1 | R | Return the CDMA module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0109 | TermWIFIExist | an 1 | R | Return the WIFI module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0110 | TermRFExist | an 1 | R | Return the RF module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0111 | TermICExist | an 1 | R | Return the IC module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0112 | TermMAGExist | an 1 | R | Return the MAG module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0113 | TermTILTExist | an 1 | R | Return the TILT module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0114 | TermWCDMAExist | an 1 | R | Return the WCDMA module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0115 | TermTOUCHSCRExist | an 1 | R | Return the touch screen module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0116 | TermCOLORSCRExist | an 1 | R | Return the color screen module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0117 | TermScrSize | n 8 | R | The size of the screen, first two bytes indicate the width, last two bytes indicate the height. |
| 0118 | TermAppName | ans…33 | R | The name of the application, such as “EasyLink”. |
| 0119 | TermAppVer | ans…33 | R | The version of the application, such as “1.00.00”. |
| 0120 | TermAppSOName | ans…33 | R | The name of the dynamic library of application, such as “libabc”. |
| 0121 | TermAppSOVer | ans…33 | R | The version of the dynamic library of application, such as “1.00.00”. |
| 0122 | TermPubSOName | ans…33 | R | The name of the dynamic library of public, such as “libabc”. |
| 0123 | TermPubSOVer | ans…33 | R | The version of the dynamic library of public, such as “1.00.00”. |
| 0124 | TermBatteryInfo | a 1 | R | Indicates the battery information,  “6” – external power supply, battery has already completed charging.  “5” – external power supply, and battery is charging.  “4” – power 4.  “3” – power 3.  “2” – power 2.  “1” – power 1.  “0” – power too low. |
| 0125 | TermRestFSSize | an…12 | R | The rest file system size, such as: “1024567” meams file system still has 1024567bytes. |
| 0126 | TermOSName | ans…24 | R | The operation system name, such as: “prolin”, “monitor plus”. |
| 0127 | TermOSVer | Ans…8 | R | The operation system version, such as: “2.4.31”. |
| 0128 | TermPCIExist | A 1 | R | Return the PCI module exist or not,  ‘0’ - not exist.  ‘1’ - exist. |
| 0129 | TermTime | N 14 | WR | Termianl time , BCD format :YYMMDDhhmmssWW, set time with format: YYMMDDhhmmss, and get time with format: YYMMDDhhmmssWW. |

## Appendix 4 – Application parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAG** | **Element** | **Attribute** | **Read Write** | **Description** |
| 0201 | SleepModeTimeout | n…4 | WR | Timeout in 100ms for waiting timeout. Valid value should be [0, 3000]. The value of “” or 0 means waiting till user canceled or confirmed.  Such as: 0x30 0x00 means 3000. |
| 0202 | PINEncryptionType | n 2 | WR | For PIN encryption, indicates which kind of encryption is going to adopt, valid value shall be as below:  1 – TDES  2 – DUKPT |
| 0203 | PINEncryptionKeyIdx | n 2 | WR | For PIN encryption, indicates which key is going to be used for data encryption |
| 0204 | PINBlockMode | n 2 | WR | PIN block mode,  0x00 – ISO9564 format 0.  0x01 – ISO9564 format 1.  0x02 – ISO9564 format 3.  0x03 – HK EPS format. |
| 0205 | DataEncryptionType | n 2 | WR | For Data encryption, indicates which kind of encryption is going to adopt, valid value shall be as below:  0 – no encryption  1 – TDES  2 – RSA  3 - MAC |
| 0206 | DataEncryptionKeyIdx | n 2 | WR | For Data encryption, indicates which key is going to be used for data encryption |
| 0207 | FallbackAllowFlag | N 2 | WR | Indicates whether allow fallback.  0 – Fallback not allowed  1 – Fallback allowed |
| 0208 | PANMaskStartPos | n 2 | WR | Indicates the first few number of clear digits for the masked PAN, valid value: 0 to 6.  Default value: 6 |
|  |  |  |  |  |

## Appendix 5 – Transaction parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag | Element | Attribute | **Read Write** | Description |
| 0301 | CurrentTxnType | n 2 | R | Indicates current transaction type:  0 – NONE  1 – Magnetic Swipe Card  2 – Fallback Swipe Card  3 – EMV Contact Card  4 – Contactless Card  0xFF – Unknown the card type |
| 0302 | CurrentCLSSType | n 2 | R | Indicates current contactless transaction type:  0 – KERNTYPE\_DEF 1 – KERNTYPE\_JCB 2 – KERNTYPE\_MC 3 – KERNTYPE\_VIS 4 – KERNTYPE\_PBOC 5 – KERNTYPE\_AE 6 – KERNTYPE\_ZIP  0xFF – Unknown the contactless transaction type |
| 0303 | CurrentPathType | n 2 | R | Indicates current contactless path type:  0 – CLSS\_PATH\_NORMAL  1 – CLSS\_VISA\_MSD  2 – CLSS\_VISA\_QVSDC 3 – CLSS\_VISA\_VSDC 4 – CLSS\_VISA\_CONTACT 5 – CLSS\_MC\_MAG 6 – CLSS\_MC\_MCHIP 7 – CLSS\_VISA\_WAVE2 8 – CLSS\_JCB\_WAVE2 9 – CLSS\_VISA\_MSD\_CVN17 10 – CLSS\_VISA\_MSD\_LEGACY  0xFF – Unknown the contactless path type |
| 0304 | Track1 data | ans…79 | R | Track1 data, Track1 needs 79 bytes. |
| 0305 | Track2 data | ans…37 | R | Track2 data, Track2 needs 37 bytes. |
| 0306 | Track3 data | ans…107 | R | Track3 data, Track3 needs 107 bytes. |
| 0307 | Expire date | ans 4 | R | Expire date of the card. |
| 0308 | OnlineAuthorization Result | n 2 | WR | Only use for EMV contact. Indicate the result of online authorization  0: transaction approved online  1: transaction declined online  2: connect host failed |
| 0309 | Response code | An 2 | WR | Authorisation Response Code.  Code that defines the disposition of a message |
| 0310 | Auth code | An 6 | WR | Authorisation Code.  Value generated by the authorisation authority for an approved transaction |
| 0311 | Auth data | Ans…16 | WR | Issuer Authentication Data.  Data sent to the ICC for online issuer Authentication. |
| 0312 | Auth data length | N 4 | WR | Issuer Authentication Data length. |
| 0313 | Issuer script | Ans…300 | WR | Issuer script, tag 71 + tag 72 data. |
| 0314 | Issuer script length | N 4 | WR | Issuer script length |
| 0315 | Online pin input | An 1 | WR | Online pin input, Online PIN input:  0 - no  1 - yes |
| 0316 | Pin block | Ans 8 | WR | Pin block |
| 0317 | ksn | Ans 10 | WR | ksn |
| 0318 | ICS | Ans…128 | WR | Used to choose ICS. Fill in the ICS type. |
| 0319 | MaskedPAN | Ans…32 | R | Mask PAN. |
| 031A | Holder Verification Method | B1 | R | Holder Verification Method  0x00 - NO CVM  0x01 - SIGNATURE  0x02 - ONLINE PIN  0x03 - OFFLINE PIN  0x04 - REFERENCE THE CUSTOMER DEVICE |
|  |  |  |  |  |

## Appendix 6 – Data to be set before StartTransaction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAG** | **Field Name** | **Required** | **Attribute** | **Description** |
| 9F02 | TxnAmount | M | N…12 | Transaction Amount |
| 9C | TransactionType | M | Hex 2 | Transaction type |
| 5F2A | TxnCurcyCode | M | Hex 4 | Transaction Currency Code |
| 5F36 | TxnCurcyExp | M | Hex 2 | Transaction Currency Exponent |
| 9A | TxnDate | M | N 6 | Transaction Date (YYMMDD) |
| 9F21 | TxnTime | M | N 6 | Transaction Time (hhmmss) |

## Appendix 7 – Data to be set before CompleteTransaction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TAG** | **Field Name** | **Required** | **Attribute** | **Description** |
| 91 | Issuer Authentication Data | C | ans…1024 | Issuer authentication data which is contained in the host response message. (value of tag 8A) |
| 71 | Issuer Script Data 1 | C | ans…1024 | Issuer Script Data which is contained in the host response message. (value of tag 71) |
| 72 | Issuer Script Data 2 | C | ans…1024 | Issuer Script Data which is contained in the host response message. (value of tag 72) |
| 0308 | Online Authorization Result | M | N 1 | Only use for EMV contact. Indicate the result of online authorization  0: transaction approved online  1: transaction declined online  2: connect host failed |
| 8A | Response Code | M | ans2 | Only use for EMV contact. Host authorization response code |
| 89 | Authorization Code | M | ans6 | Only use for EMV contact. Host authorization code |

## Appendix 8 - EMV TAGs

### EMV TAGs can be accessed after StartTransaction:

| Tag Identifier | Description |
| --- | --- |
| 4F | Application Identifier |
| 9F12 | Application preferred Name |
| 50 | Application Label |
| 5F30 | Service Code |
| 5F20 | Cardholder Name |
| 57 | Track 2 Equivalent  Primary Account Number  Field Separator (Hex 'D')  Expiration Date (YYMM)  Service Code  Discretionary Data (defined by individual payment systems)  Pad with one Hex 'F' if needed to ensure whole bytes |
| 5A | Application Primary Account Number |
| 5F24 | Application Expiry Date |
| 5F34 | Application Primary account Number sequence number |
| 5F25 | Application Effective Date |
| 95 | Terminal Verification Results |
| 9B | Transaction Status Information |
| 9F27 | Cryptogram Information Data |
| 8A | Authorization response Code |
| 9F26 | Cryptogram returned by the ICC in response of the GENERATE AC command |
| 9F36 | Application Transaction Counter  Counter maintained by the application in the ICC (incrementing the ATC is managed by the ICC) |
| 9F10 | Issuer Application Data |
| 9F41 | Transaction Sequence Counter from terminal |
| 9F02 | Amount Authorized |
| 9F03 | Amount (other) (TIP) |
| 5F36 | Transaction currency component |
| 9F1B | Terminal Floor Limit |
| 9F1C | Terminal Identification |
| 9F35 | Terminal Type |
| 9F1A | Terminal Country Code |
| 5F2A | Transaction Currency Code |
| 82 | Application Interchange Profile  Indicates the capabilities of the card to support specific functions in the application |
| 9F37 | Unpredictable Number  Value to provide variability and uniqueness to the generation of a cryptogram |
| 9C | Transaction Type |
| 84 | Dedicated File Name |
| 9F09 | Application Version Number |
| 9F34 | CVM Cardholder verification Method Result |
| 9F07 | Application Usage Control  Indicates issuer‘s specified restrictions on the geographic usage and services allowed for the application |
| 9F0D | Issuer Action Code-Default |
| 9F0E | Issuer Action Code – Denial |
| 9F0F | Issuer Action Code – Online |
| 9F33 | Terminal Capabilities |
| 5F34 | Application PAN sequence Number |
| 9F39 | POS Entry Mode |
| 8E | CVM List  Identifies a method of verification of the cardholder supported by the application |
| 5A | Application Primary Account Number |
| 57 | Track 2 equivalent |
| 5F20 OR 9F0B | Cardholder name or Extended |
| 9F1F | Track 2 discretionary data |
| 9F7A | VLP process indicator |
| 9F74 | VLP authorization code |

### CLSS TAGs can be accessed after StartTransaction

…..

### EMV TAGs can be accessed after CompleteTransaction

### CLSS TAGs can be accessed after CompleteTransaction

## Appendix 9 – Value attribute

Below are the EMV standard data formats:

|  |  |
| --- | --- |
| a | Alphabetic data elements contain a single   character per byte. The permitted characters are alphabetic only (a to z and   A to Z, upper and lower case). |
| an | Alphanumeric data elements contain a single   character per byte. The permitted characters are alphabetic (a to z and A to   Z, upper and lower case) and numeric (0 to 9). |
| ans | Alphanumeric Special data elements contain   a single character per byte. The permitted characters and their coding are   shown in the Common Character Set table in Annex B of Book 4. There is one   exception: The permitted characters for Application Preferred Name are the   non-control characters defined in the ISO/IEC 8859 part designated in the   Issuer Code Table Index associated with the Application Preferred Name. |
| b | These data elements consist of either   unsigned binary numbers or bit combinations that are defined elsewhere in the   specification. Binary example: The Application Transaction Counter (ATC) is   defined as ―b‖ with a length of two   bytes. An ATC value of 19 is stored as Hex '00 13'. |
| cn | Compressed numeric data elements consist of   two numeric digits (having values in the range Hex '0'–'9') per byte. These   data elements are left justified and padded with trailing hexadecimal 'F's.   Example: The Application Primary Account Number (PAN) is defined as ―cn‖ with a length of up to ten bytes. A value   of 1234567890123 may be stored in the Application PAN as Hex '12 34 56 78 90   12 3F FF' with a length of 8. |
| n | Numeric data elements consist of two   numeric digits (having values in the range Hex '0' – '9') per byte. These   digits are right justified and padded with leading hexadecimal zeroes. Other   specifications sometimes refer to this data format as Binary Coded Decimal   (―BCD‖) or unsigned packed.   Example: Amount, Authorised (Numeric) is defined as ―n 12‖ with a length of six bytes. A value of   12345 is stored in Amount, Authorised (Numeric) as Hex '00 00 00 01 23 45' |
| var. | Variable data elements are variable length   and may contain any bit combination. Additional information on the formats of   specific variable data elements is available elsewhere. |

## Appendix 10 – File type description for file download

Terminal supports file with below suffix in <FileDownload> API:

1) .ui : for UI layout file (XML format);

2) .emv: for EMV contact parameter file (XML format);

3) .clss: for EMV contactless parameter file (XML format);

4) .font: for font file (binary format);

5) .so: only for library file to the terminal with Prolin platform, like d220;

Note: D180S is not support this kind of file;

6) .bmp, .png, .gif: for picture file;

Note: D180 only support .bmp format file;

~~7) .bin: for application or monitor file to the terminal with Monitor platform;~~

7) .bin: for application file to the terminal with monitor platform;

8) .monitor: for monitor file to the terminal with monitor platform;

9) .aip: for application file to the terminal with Prolin platform;

10) .lng: for translation file to the terminal with Prolin or Monitor platform;

11) .os: for Prolin OS file;

**Attention**:

1. If the name and suffix of the file to be downloaded is already exists in the terminal, then the new file shall overwrite the file in the terminal;

2. If the file to be downloaded is application, monitor, Prolin OS, or font file, then the terminal shall reboot automatically after download successful;