

# **LANDI Light EMV Application Programming Guide**

EMV Card Payment Application Handy Development

2017/12/27

**LANDI**

Version: 2.6

# Contents

Contents .....	1
1. Summarize .....	3
1.1 Introduction.....	3
1.2 Audience.....	3
1.3 Coding Scheme.....	3
1.4 Supporting Transaction Folw.....	4
2. Application Module Architecture.....	5
3. Transaction Time Sequence.....	6
4. Event and Signal .....	7
5. Transaction Flow Diagrams.....	9
5.1 Kernel Initialization.....	11
5.2 EMV Contact Level2.....	12
5.3 PBOC ECash .....	14
5.4 PBOC Contactless Level2.....	16
5.5 PBOC Quick Pay (qPBOC) .....	18
5.6 VISA (PayWave) .....	20
5.7 Master Card(PayPass).....	23
5.8 AMEX .....	26
5.9 DISCOVER.....	29
5.10 JCB.....	32
5.11 Query ICC Transaction log.....	34
5.12 Query ICC Offline Balance.....	35
5.13 Transaction Finish Process .....	36
6. API Specification .....	37
6.1 EMV_Kern_uiCreateObject.....	37
6.2EMV_Kern_uiDestroyObject.....	38
6.3EMV_Kern_uiManageAID .....	39
6.4EMV_Kern_uiUpdateCAIndexList.....	40
6.5EMV_Kern_uiSetCAPubKey .....	41
6.6EMV_Kern_uiSetCAPubKey_SM .....	42
6.7EMV_Kern_uiManageRecCert.....	43
6.8 EMV_Kern_uiManageDOL.....	44
6.9 EMV_Kern_uiSignalInTLV .....	45
6.10 EMV_Kern_uiSetTLV.....	47
6.11 EMV_Kern_uiSetTLVList .....	48
6.12 EMV_Kern_uiGetTLV .....	51
6.13 EMV_kern_uiGetBalance.....	52
6.14 EMV_kern_uiGetDataAPDU .....	53
6.15 EMV_Kern_uiGetICCLog.....	54
6.16 EMV_Kern_uiGetECCLog .....	55
6.17 EMV_Kern_vSwitchDebug .....	56

6.18 EMV_kern_uiSetHandle	57
7. Structure Definition	58
EMV_Configuration	58
EMV_tPKFILESTRU	58
EMV_tPKFILESTRU_SM	59
EMV_tSelectAID	60
EMV_tRecCert	61
EMV_tICCLog	62
EMV_tECCLog	63
EMV_tCandAIDInfo	64
EMV_tAIDCandList	65
EMV_tFinalData	66
EMV_tRecordData	67
EMV_tCVM	68
EMV_tDisplayMsg	69
EMV_tErrorID	70
EMV_tTransData	71
EMV_EXPAND_BASEFUN	72
EMV_EXPAND_INTERFACE	73
8. Obtain Kernel Debug log	75
Annex A TAG Dictionary	76
A.1 TAG of EMV_KERNELID_EMV	76
A.2 TAG of EMV_KERNELID_PBOC	79
A.3 TAG of EMV_KERNELID_VISA	81
A.4 TAG of EMV_KERNELID_MASTER	83
A.5 TAG of EMV_KERNELID_AMEX	84
A.6 TAG of EMV_KERNELID_DISCOVER	85
A.7 TAG of EMV_KERNELID_JCB	86
A.8 TAG of EMV_KERNELID_DEFINE	88
Annex B Macro Definition	95
Annex C Transaction Return Code	99

# 1. Summarize

## 1.1 Introduction

This document introduces an EMV library which covers various kinds of EMV transaction flow and their complicated logic process. It is easy for an application developer to program a payment application and make the application easy to maintain. It is an easier, faster and flexible EMV library.

## 1.2 Audience

This specification is intended for use by EMV payment application developer.

## 1.3 Coding Scheme

BCD

### Binary Coded Decimal

BCD encoded Amount:

For example: 6 Byte Authorized Amount: 123.45, BCD coding as: \x00\x00\x00\x01\x23\x45  
(Amount should be right most)

BCD encoded Date:

For example: March 10<sup>th</sup> 2015, BCD coding as: \x20\x15\x03\x10

BCD encoded Time:

For example: 08:10:59, BCD coding as: \x08\x10\x59

HEX

### Hexadecimal coding

HEX encoded AID:

For example: AID: A000000333010101, HEX coding as: \xA0\x00\x00\x03\x33\x01\x01

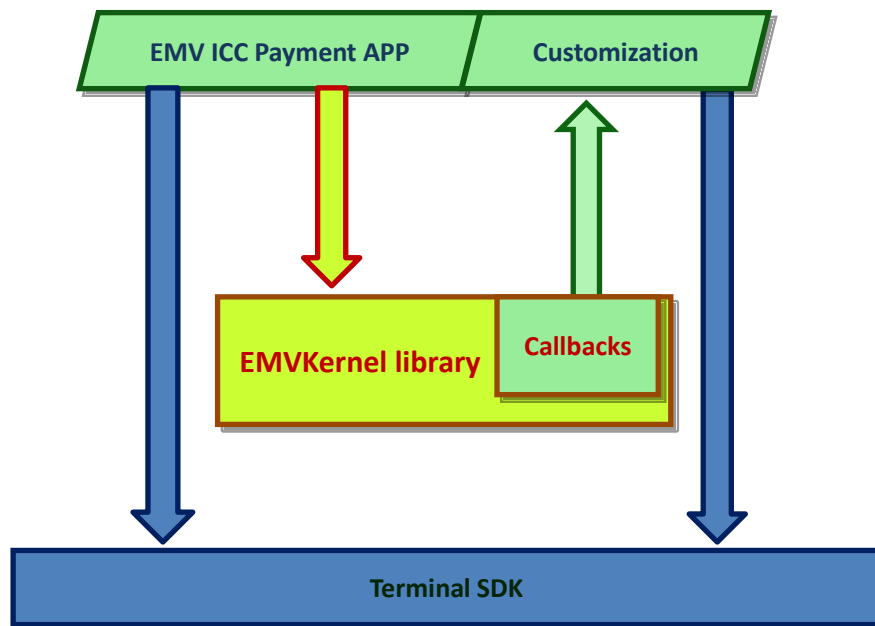
## **1.4 Supporting Transaction Folw**

- EMV Contact Level2 4.3
- VISA PayWave 2.1.3(qVSDC)/2.2
- MasterCard PayPass 3.0
- PBOC3.0 Contact Level2
- PBOC3.0 Contactless Level2
- PBOC3.0 ECash
- PBOC3.0 Quick Pay(qPBOC)
- American Express ExpressPay 3.1(AMEX3.1)
- Discover® Contactless D-PAS v1.1 & ZIP Payment v3.1.2(DISCOVER)

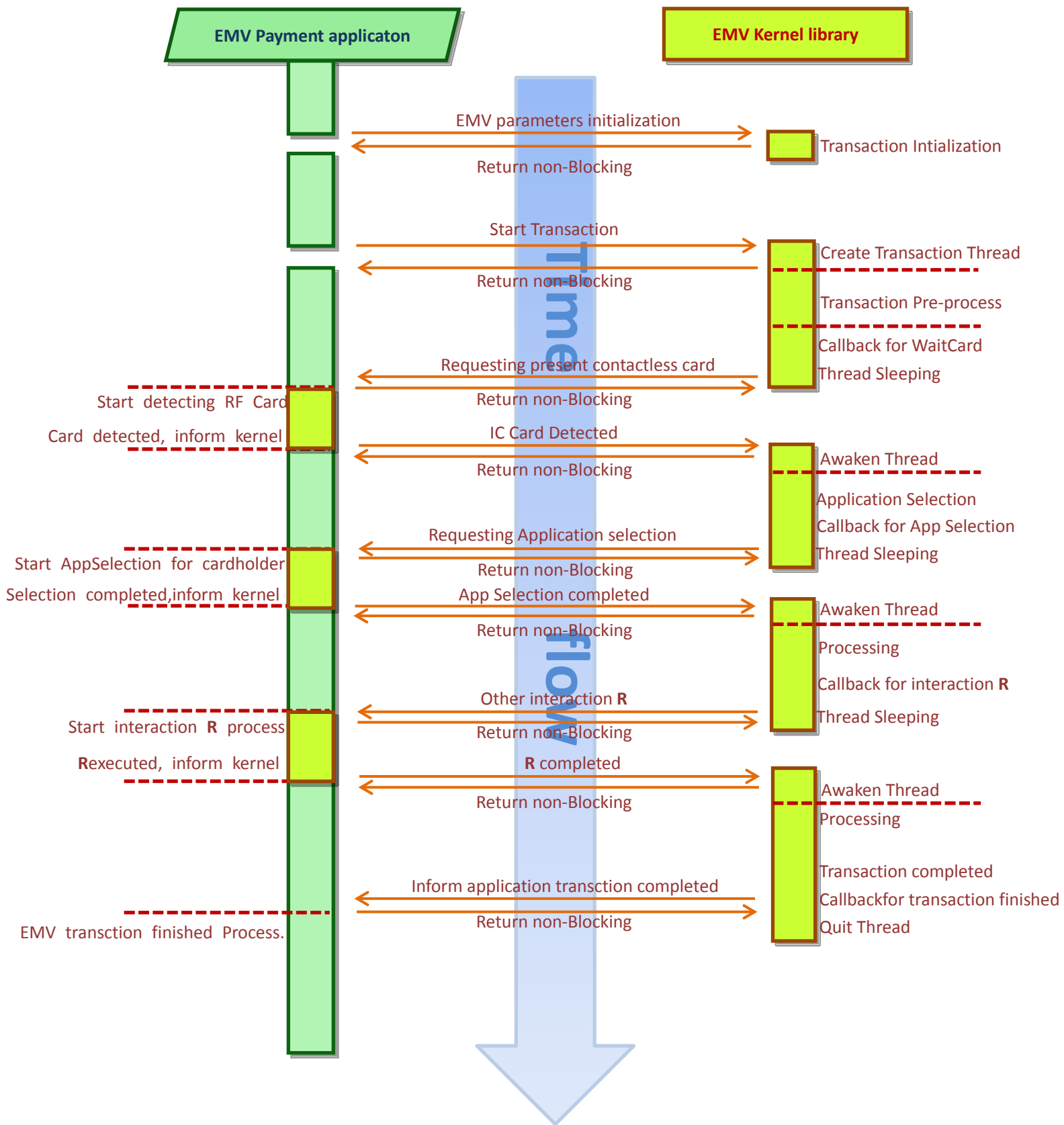
## 2. Application Module Architecture

The SDK APIs provide chipcard/contactless device I/O and APDU APIs such as open/close device and send APDU command.

The EMV Kernel is a library based on SDK, which provides APIs to process many kinds of transaction flow, such as EMV, VISA, Master Card, PBOC, qPBOC, AMEX3.1, DISCOVER.



### 3. Transaction Time Sequence



## 4. Event and Signal

The following table lists all of the events and signal parameters when event process complete.

**Table 4-1 Event and Signal**

\*Note: (M)-Mandatory (O)-Option (C)-Condition

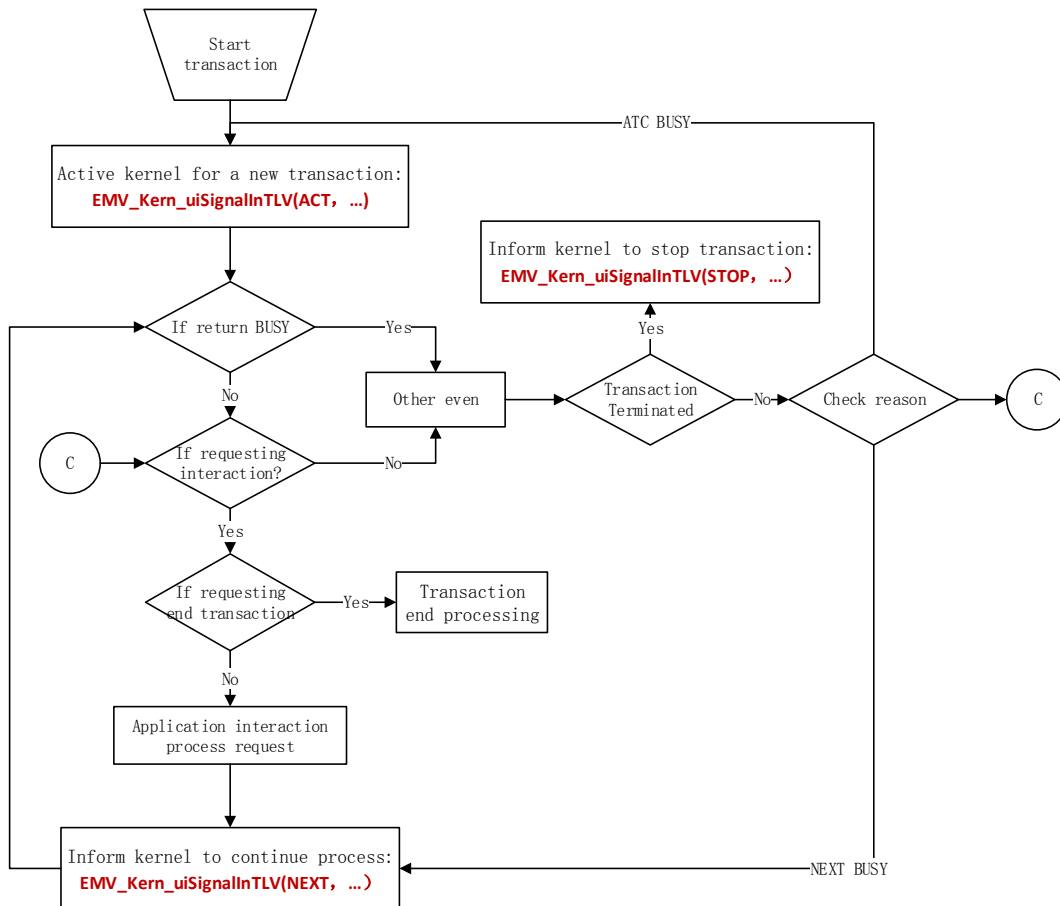
Events	Callback Function	Request Description	Wait Signal	Signal Response
		Activate a new transaction starting from application selection.	EMV_SIGNAL_ACT	DEF_TAG_PSE_FLAG (O) DEF_TAG_START_RECOVERY (O) DEF_TAG_QUERY_ICCLOG (O)
Wait contactless Card	EXEP_ucWaitCard	Requesting cardholder to show contaless card.	EMV_SIGNAL_NEXT	NULL
Application Selection	EXEP_ucAppSelection	Requesting cardholder to select IC card applicaton.	EMV_SIGNAL_NEXT	EMV_TAG_TM_AID (M)
After Final Selection	EXEP_ucFinalSit	Supply a time slot for application to adjust parameter according finial selected AID.	EMV_SIGNAL_NEXT	Please refer to this document flow chart of part 5 transactions in the request parameter table of all kinds of trading process
After Read Record	EXEP_ucReadRecord	Supply a time slot for application to deal with messages from card records.	EMV_SIGNAL_NEXT	DEF_TAG_PAN_IN_BLACK (O) DEF_TAG_ACCUMULATE_AMOUNT (O)
Cardholder Verification	EXEP_ucCardHolderVerify	Requesting cardholder to made a verification according CVM which indicated by EMV kernel.	EMV_SIGNAL_NEXT	DEF_TAG_CHV_STATUS (M)
Online Authorization	EXEP_ucOnlineProcess	Requesting online authorization.	EMV_SIGNAL_NEXT	DEF_TAG_ONLINE_STATUS (M) If online communication is success, following is necessary while retured by host service. EMV_TAG_TM_ARC (C) DEF_TAG_AUTHORIZE_FLAG (C) EMV_TAG_TM_AUTHCODE (C) DEF_TAG_HOST_TLVDATA (C)
Transaction Finished	EXEP_vEndProcess	Inform that transaction is finished.		
Kernel Obtain	EXEP_vObtain	Obtain some messages from application by EMV kernel. Such as torn transaction	EMV_SIGNAL_NEXT	



		information.		
<b>Kernel Send Out</b>	<b>EXEP_vSendOut</b>	Send out some messages from EMV kernel such as torn transaction information、inform message.		
		Stop or terminate a transaction after transaction activated but haven't finished.	EMV_SIGNAL_STOP	

## 5. Transaction Flow Diagrams

EMV transaction is running based on event and signal mechanism. After activating a transaction, EMV kernel will process and conduct the procedures till transaction is completed. During transaction process, EMV kernel will generate one or multiple events for interaction when necessary. These events have different purposes, including cardholder application selection, cardholder verification or transaction related parameters setting, etc. These events are generated depending on transaction related parameters and IC card logic design, and it is unpredictable. Event is generated based on callback mechanism. After generating events, EMV kernel will go into sleeping and waiting for the signal to awake after finishing event process by application. After EMV kernel awakes, EMV kernel will proceed until transaction finishes event generation.



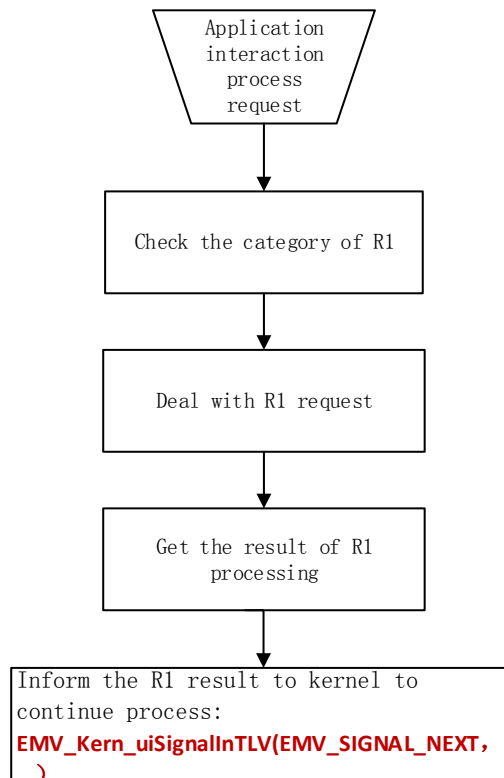
For example, when EMV Kernel generating an event marking with R1, the Implement of callback function should be simple and follow these steps:

- 1、Store the parameters which return with event R1.
- 2、Set the event flag with R1.
- 3、Return success.

**Notice:**

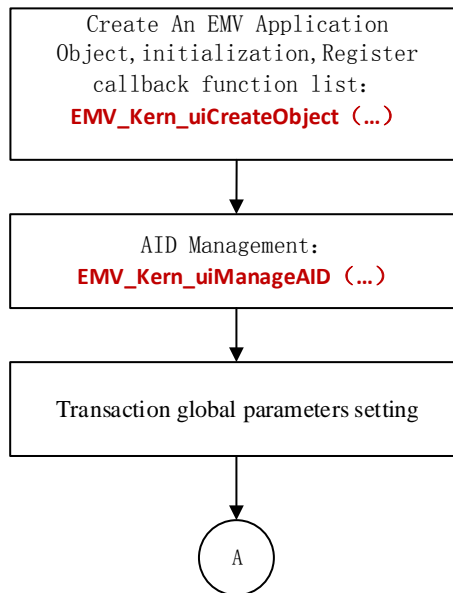
**Never call the API `EMV_Kern_uiSignalInTLV` in callback function implementation.**

**Reference to the example of `EMV_Kern_uiCreateObject`**



## 5.1 Kernel Initialization

Before activating any transaction, EMV kernel initialization is necessary and mandatory. The initialization should be executed once after emv payment application starting. After kernel initialization, application is allowed to set these parameters which are global and always with same values for difference transaction.



## 5.2 EMV Contact Level2

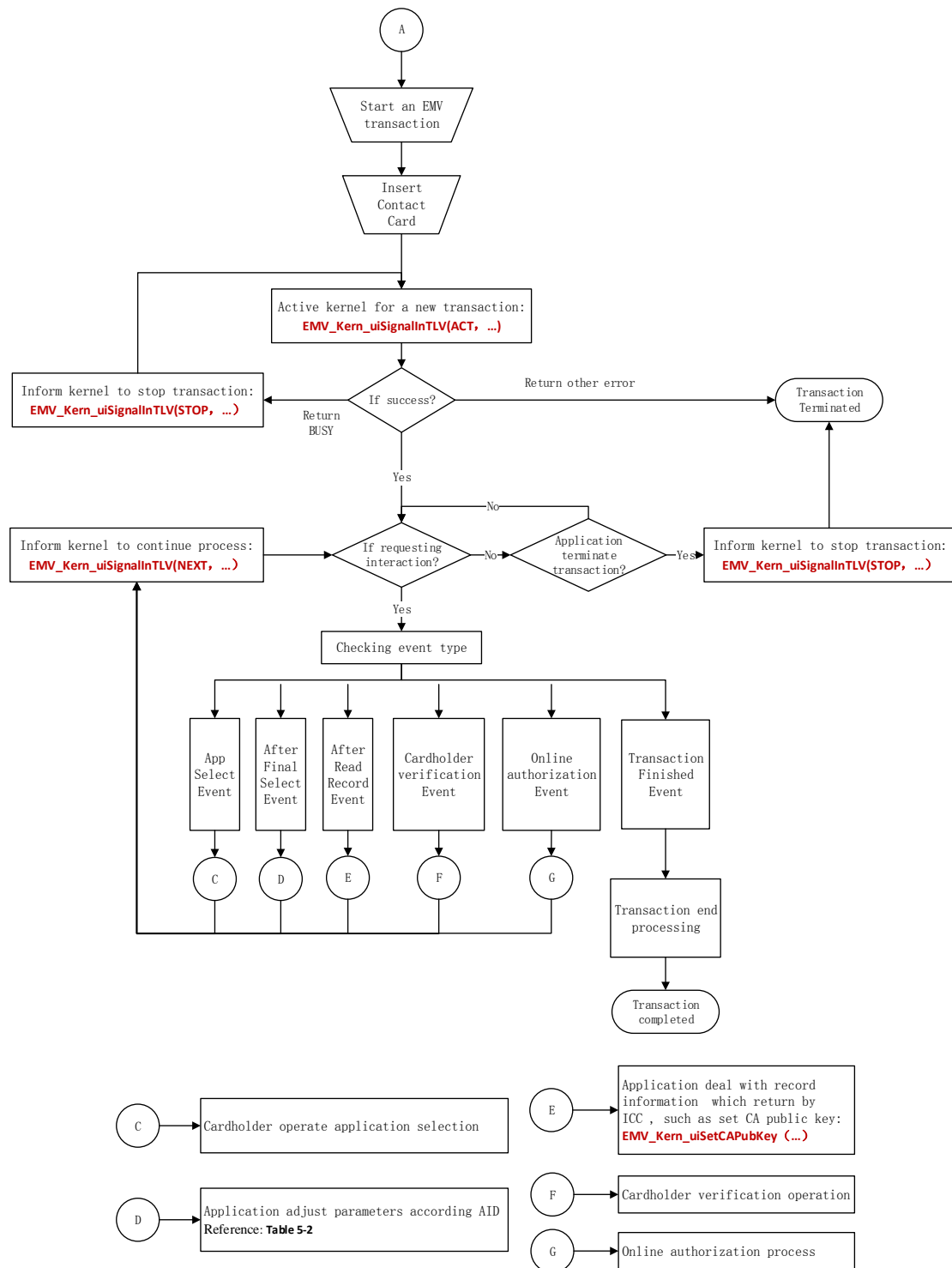


Table 5-2 Transaction Parameter requirement for contact EMV L2

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for contact EMV L2 flow:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN (M) REF_V:0x00000000100(1.00)</li> <li>• EMV_TAG_TM_OTHERAMNTN (O) REF_V:0x000000000000</li> <li>• EMV_TAG_TM_TRANSDATE (M)REF_V:0x171216</li> <li>• EMV_TAG_TM_TRANSTIME (M)REF_V:0x131535</li> <li>• EMV_TAG_TM_TRSEQCNTR (M)REF_V:0x00001234</li> <li>• EMV_TAG_TM_TERMTYPE (M)REF_V:0x22</li> <li>• EMV_TAG_TM_CAP (M)REF_V:0xE0F8C8</li> <li>• EMV_TAG_TM_CAP_AD (M) REF_V: 0x6000F0A001</li> <li>• EMV_TAG_TM_CNTRYCODE (M) REF_V:0x0156</li> <li>• EMV_TAG_TM_CURCODE (M) REF_V:0x0156</li> <li>• EMV_TAG_TM_TRANSTYPE (M)REF_V:0x00(Purchase)</li> <li>• EMV_TAG_TM_FLOORLMT (O) REF_V:0x00002710(100.00)</li> <li>• DEF_TAG_TAC_DECLINE (O) REF_V:0x0000000000</li> <li>• DEF_TAG_TAC_ONLINE (O) REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_TAC_DEFAULT (O) REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_GAC_CONTROL (O) REF_V:0x00</li> <li>• DEF_TAG_SERVICE_TYPE (M) REF_V:0x00(Purchase)</li> <li>• DEF_TAG_RAND_SLT_THRESHOLD (M) REF_V:0x000000001000(10)</li> <li>• DEF_TAG_RAND_SLT_PER (M) REF_V:0x30</li> <li>• DEF_TAG_RAND_SLT_MAXPER (M) REF_V:0x90</li> <li>• C_TAG_TM_DF69 (O)REF_V:0x01</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_EMV</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_EMV</a>)</p>

## 5.3 PBOC ECash

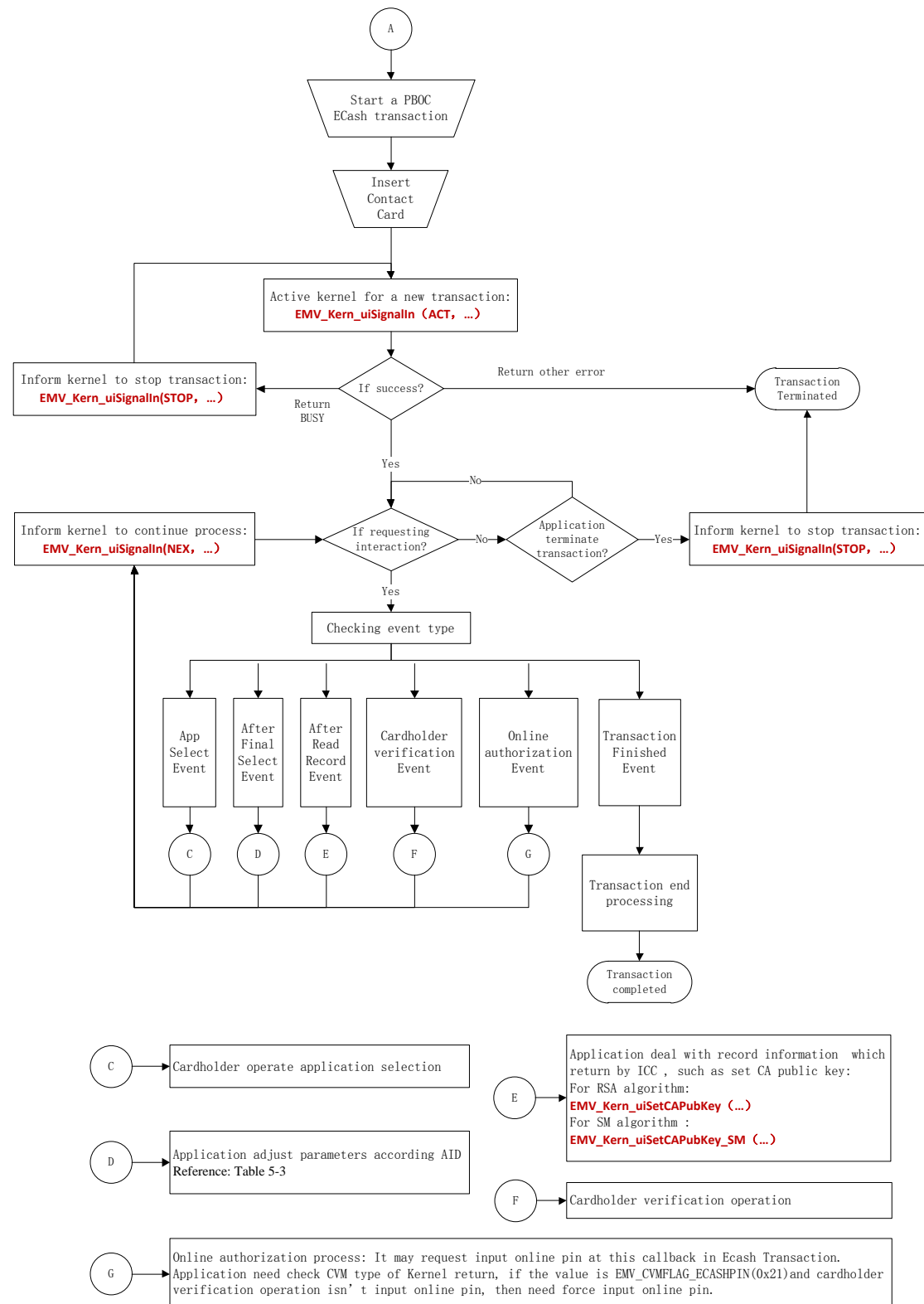


Table 5-3 Transaction Parameter requirement for PBOC ECash

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for PBOC ECash:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN (M)REF_V:0x00000000100(1.00)</li> <li>• EMV_TAG_TM_OTHERAMNTN (O)REF_V:0x000000000000</li> <li>• EMV_TAG_TM_TRANSDATE (M)REF_V:0x171216</li> <li>• EMV_TAG_TM_TRANSTIME (M)REF_V:0x131535</li> <li>• EMV_TAG_TM_TRSEQCNTR (M)REF_V:0x00001234</li> <li>• EMV_TAG_TM_TERMTYPE (M)REF_V:0x22</li> <li>• EMV_TAG_TM_CAP (M)REF_V: 0xE0F8C8</li> <li>• EMV_TAG_TM_CAP_AD (M)REF_V: 0x6000F0A001</li> <li>• EMV_TAG_TM_CNTRYCODE (M)REF_V:0x0156</li> <li>• EMV_TAG_TM_CURCODE (M)REF_V:0x0156</li> <li>• EMV_TAG_TM_TRANSTYPE (M)REF_V:0x00(Purchase)</li> <li>• EMV_TAG_TM_FLOORLMT (O)REF_V:0x00002710(100.00)</li> <li>• DEF_TAG_TAC_DECLINE (O)REF_V:0x0000000000</li> <li>• DEF_TAG_TAC_ONLINE (O)REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_TAC_DEFAULT (O)REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_SERVICE_TYPE (M)REF_V:0x00(Purchase)</li> <li>• C_TAG_TM_9F7A (M)REF_V:0x01</li> <li>• C_TAG_TM_DF69 (O)REF_V:0x01</li> <li>• C_TAG_TM_9F7B (M)REF_V:0x000000010000(100.00)</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_PBOC</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID =<a href="#">EMV_KERNELID_PBOC</a>)</p>



## 5.4 PBOC Contactless Level2

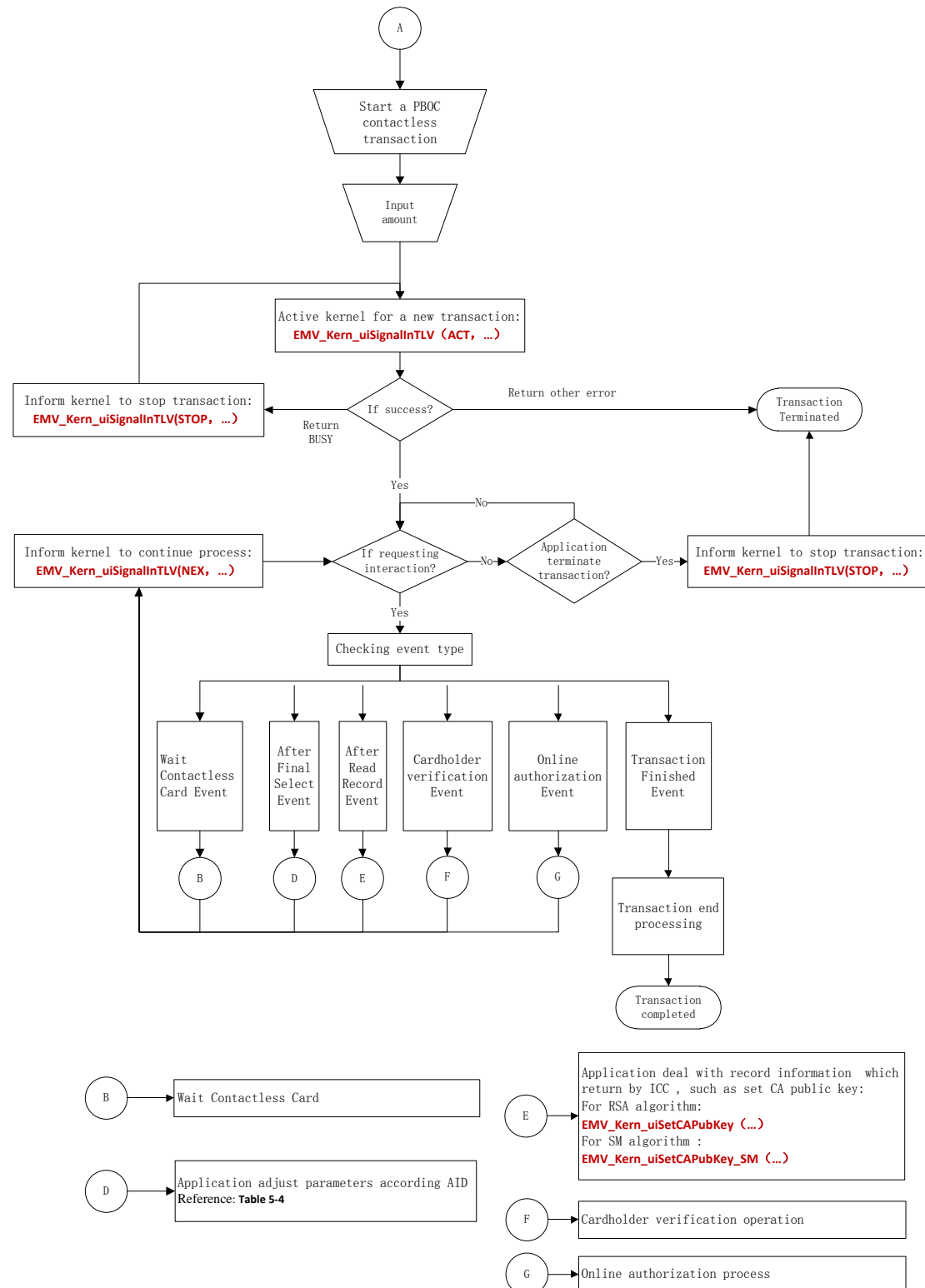


Table 5-4 Transaction Parameter requirement for PBOC CTLess L2

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for PBOC contactless flow:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN (M) REF_V:0x00000000100(1.00)</li> <li>• EMV_TAG_TM_OTHERAMNTN (O) REF_V:0x000000000000</li> <li>• EMV_TAG_TM_TRANSDATE (M)REF_V:0x171216</li> <li>• EMV_TAG_TM_TRANSTIME (M)REF_V:0x131535</li> <li>• EMV_TAG_TM_TRSEQCNTR (M)REF_V:0x00001234</li> <li>• EMV_TAG_TM_TERMTYPE (M)REF_V:0x22</li> <li>• EMV_TAG_TM_CAP (M)REF_V: 0xE0F8C8</li> <li>• EMV_TAG_TM_CAP_AD (M) REF_V: 0x6000F0A001</li> <li>• EMV_TAG_TM_CNTRYCODE (M) REF_V:0x0156</li> <li>• EMV_TAG_TM_CURCODE (M) REF_V:0x0156</li> <li>• EMV_TAG_TM_TRANSTYPE (M)REF_V:0x00(Purchase)</li> <li>• EMV_TAG_TM_FLOORLMT (O) REF_V:0x00002710(100.00)</li> <li>• DEF_TAG_TAC_DECLINE (O) REF_V:0x0000000000</li> <li>• DEF_TAG_TAC_ONLINE (O) REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_TAC_DEFAULT (O) REF_V:0xFFFFFFFF</li> <li>• DEF_TAG_GAC_CONTROL (O) REF_V:0x00</li> <li>• DEF_TAG_SERVICE_TYPE (M) REF_V:0x00(Purchase)</li> <li>• DEF_TAG_RAND_SLT_THRESHOLD (M) REF_V:0x000000001000(10)</li> <li>• DEF_TAG_RAND_SLT_PER (M) REF_V:0x30</li> <li>• DEF_TAG_RAND_SLT_MAXPER (M) REF_V:0x90</li> <li>• C_TAG_TM_9F66 (M)REF_V:0x67004080</li> <li>• C_TAG_TM_DF69 (O)REF_V:0x01</li> <li>• C_TAG_TM_TRANS_LIMIT (O)REF_V:0x0000000010000(100.00)</li> <li>• C_TAG_TM_CVM_LIMIT (O)REF_V:0x0000000010000(100.00)</li> <li>• C_TAG_TM_FLOOR_LIMIT (O)REF_V:0x0000000010000(100.00)</li> <li>• C_TAG_TM_RD_RCP (O)REF_V:0x7C00</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_PBOC</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID =<a href="#">EMV_KERNELID_PBOC</a>)</p>

## 5.5 PBOC Quick Pay (qPBOC)

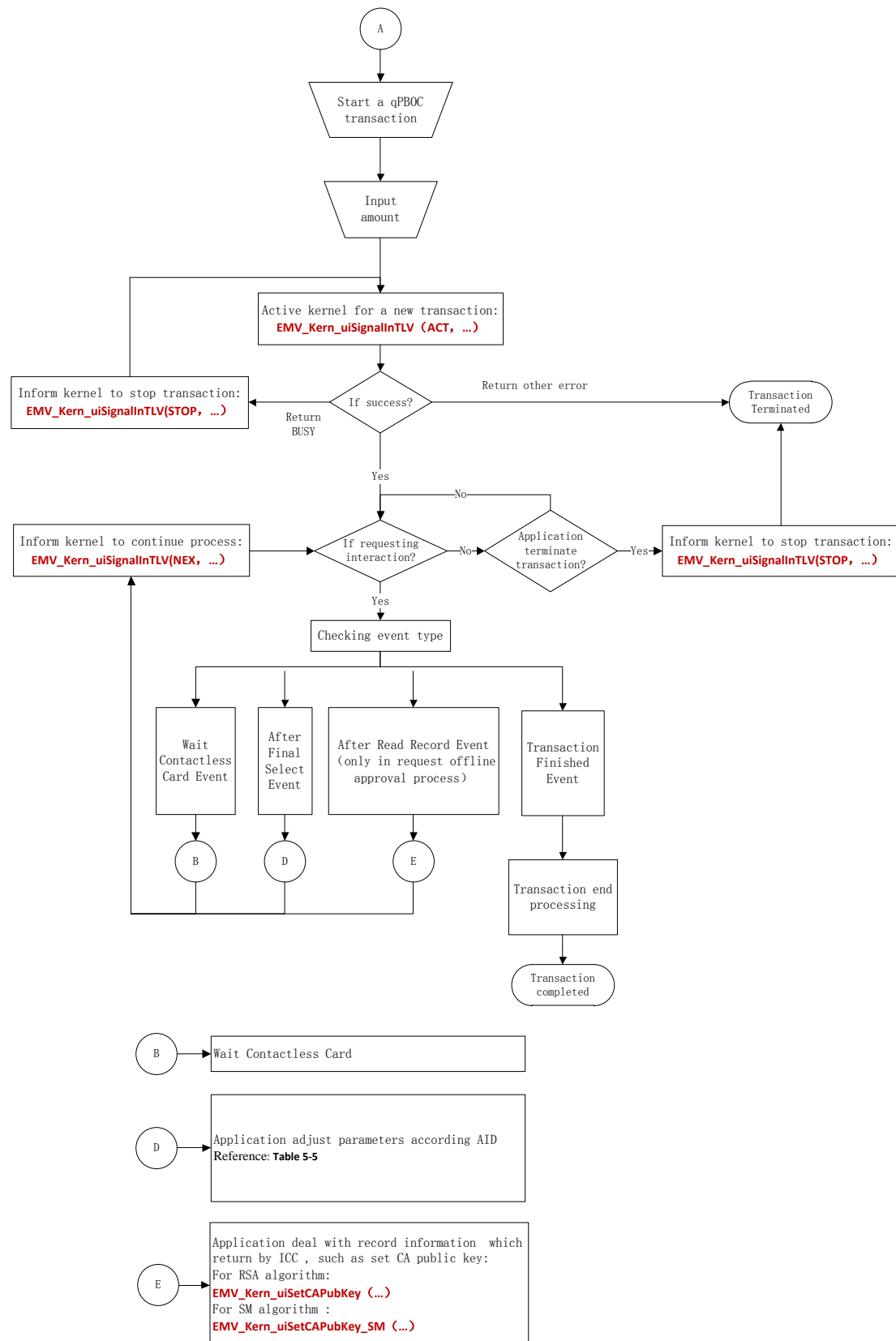


Table 5-5 Transaction Parameter requirement for qPBOC

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for qPBOC flow:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN (M)REF_V:0x00000000100(1.00)</li> <li>• EMV_TAG_TM_OTHERAMNTN (O)REF_V:0x000000000000</li> <li>• EMV_TAG_TM_TRANSDATE (M)REF_V:0x171216</li> <li>• EMV_TAG_TM_TRANSTIME (M)REF_V:0x131535</li> <li>• EMV_TAG_TM_TRSEQCNTR (M)REF_V:0x00001234</li> <li>• EMV_TAG_TM_CURCODE (O)REF_V:0x0156</li> <li>• EMV_TAG_TM_TRANSTYPE (O)REF_V:0x00(Purchase)</li> <li>• C_TAG_TM_9F66 (M)REF_V:0x27004080</li> <li>• C_TAG_TM_DF69 (O)REF_V:0x01</li> <li>• C_TAG_TM_TRANS_LIMIT (O)REF_V:0x000000010000(100.00)</li> <li>• C_TAG_TM_CVM_LIMIT (O)REF_V:0x000000010000(100.00)</li> <li>• C_TAG_TM_FLOOR_LIMIT (O)REF_V:0x000000010000(100.00)</li> <li>• DEF_TAG_TORN_SUPPORT (O)REF_V:0x01</li> <li>• DEF_TAG_ALLOW_DUP_ICC_SAMEVALUE (O)REF_V:0x01</li> <li>• C_TAG_TM_RD_RCP (O)REF_V:0x7C00</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_PBOC</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_PBOC</a>)</p>

## 5.6 VISA (PayWave)

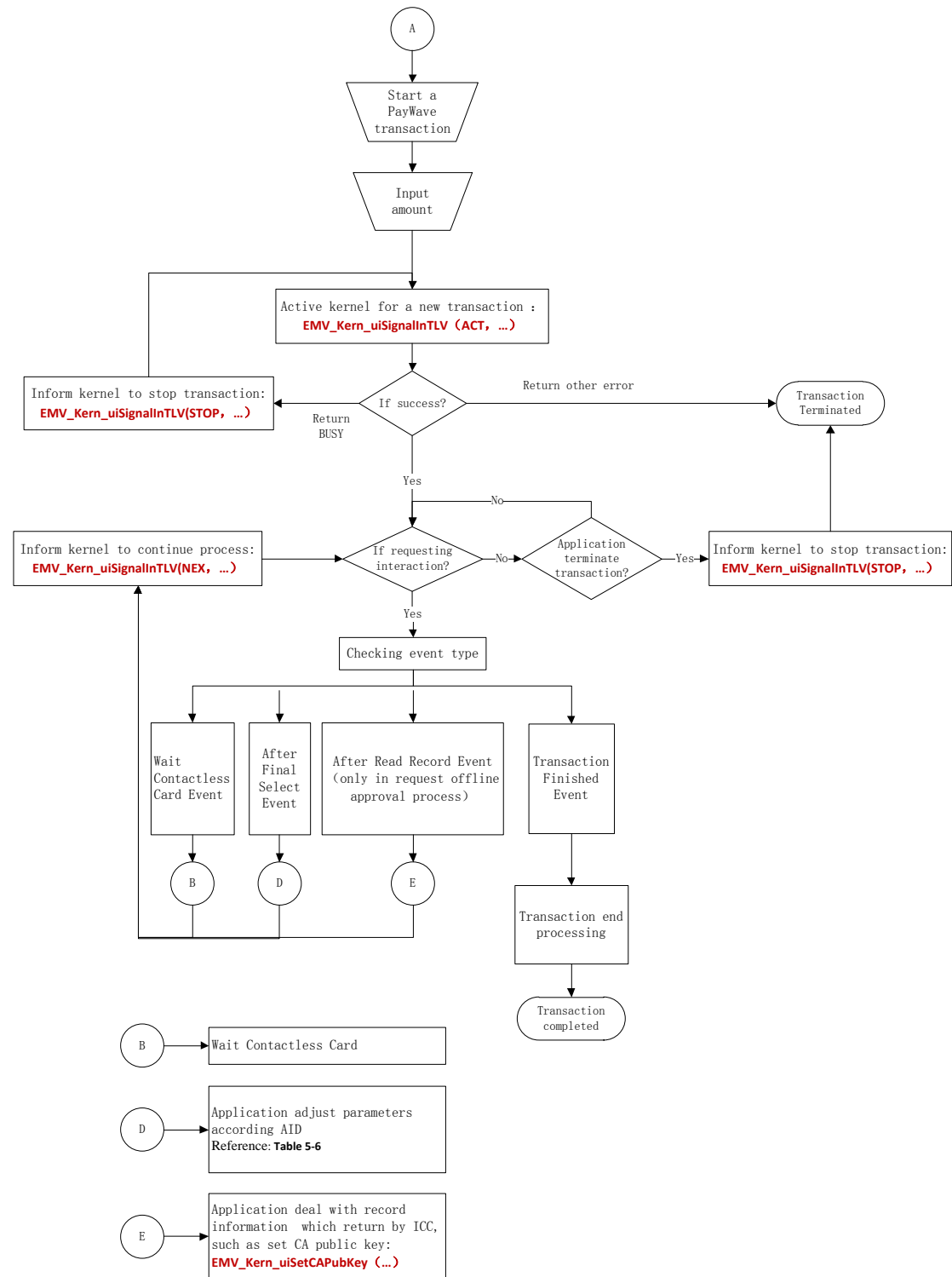


Table 5-6 Transaction Parameter requirement for PayWave

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>ACT: active kernel of a new transaction</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. All parameters Tag are required to set at this step as bellow:</p> <ul style="list-style-type: none"> <li>EMV_TAG_TM_TRANSTYPE(O) REF_V:0x00(Purchase)</li> <li>EMV_TAG_TM_AUTHAMNTN(M) REF_V:0x00000000100(1.00)</li> <li>EMV_TAG_TM_OTHERAMNTN(O) REF_V:0x0000000000</li> <li>EMV_TAG_TM_TRANSDATE(M) REF_V:0x171216</li> <li>EMV_TAG_TM_TRANSTIME(M) REF_V:0x131535</li> <li>EMV_TAG_TM_TRSEQCNTR(O) REF_V:0x00001234</li> <li>EMV_TAG_TM_TERMTYPE(O) REF_V:0x22</li> <li>EMV_TAG_TM_CNTRYCODE(M) REF_V:0x0840</li> <li>EMV_TAG_TM_CURCODE(M) REF_V:0840</li> <li>V_TAG_TM_9F66(M) REF_V:0x26004000</li> <li>V_TAG_RD_RCP(O) REF_V:0x7C00</li> <li>V_TAG_TM_TRANS_LIMIT(O) REF_V:0x000000010000(100.00)</li> <li>V_TAG_TM_FLOOR_LIMIT(O) REF_V:0x000000008000(80.00)</li> <li>V_TAG_TM_CVM_LIMIT(O) REF_V:0x000000006000(60.00)</li> <li>EMV_TAG_TM_FLOORLMT(O)<sup>1</sup> REF_V:0x00002710(100.00)</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>
<b>E: After Read Application Data</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>

Remarks:

The EMVKernel will not required an 'Online Authorization Event' when the transaction performs an Approved offline. EMVKernel will only required ONLINE\_PIN at 'Transaction End

<sup>1</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B) is 4 Bytes Hex Code, Such as \$100.00, Setting need like this "\x00\x00\x27\x10".

Event', and Application can get CVM Type from data of 'EMV\_tTransData->ucCVM' and perform CVM indicate the cardholder. And Kernel will not require an ONLINE\_PIN if transaction performed approve offline.

The detail Result Code of Transaction, you can refer the [<Annex C Transaction Return Code>](#).

The Return Code about 'EMV\_RESULT\_REPOWERICC' which return from kernel when do an Visa transaction commented as below:

In GPO Step, Kernel will return EMV\_RESULT\_REPOWERICC when card return SW=6986, and application need power off RF field period 1 to 1.5 seconds and power on it again, and prompt the cardholder the message "Refer to you device", then present the card again do the transaction to completion. During the second Present Card, the context of the current transaction persists unchanged, including Amount, Authorized and Transaction Currency code, and Pre-Condition and Pre-Preprocessing data remains unchanged.

## 5.7 Master Card(PayPass)

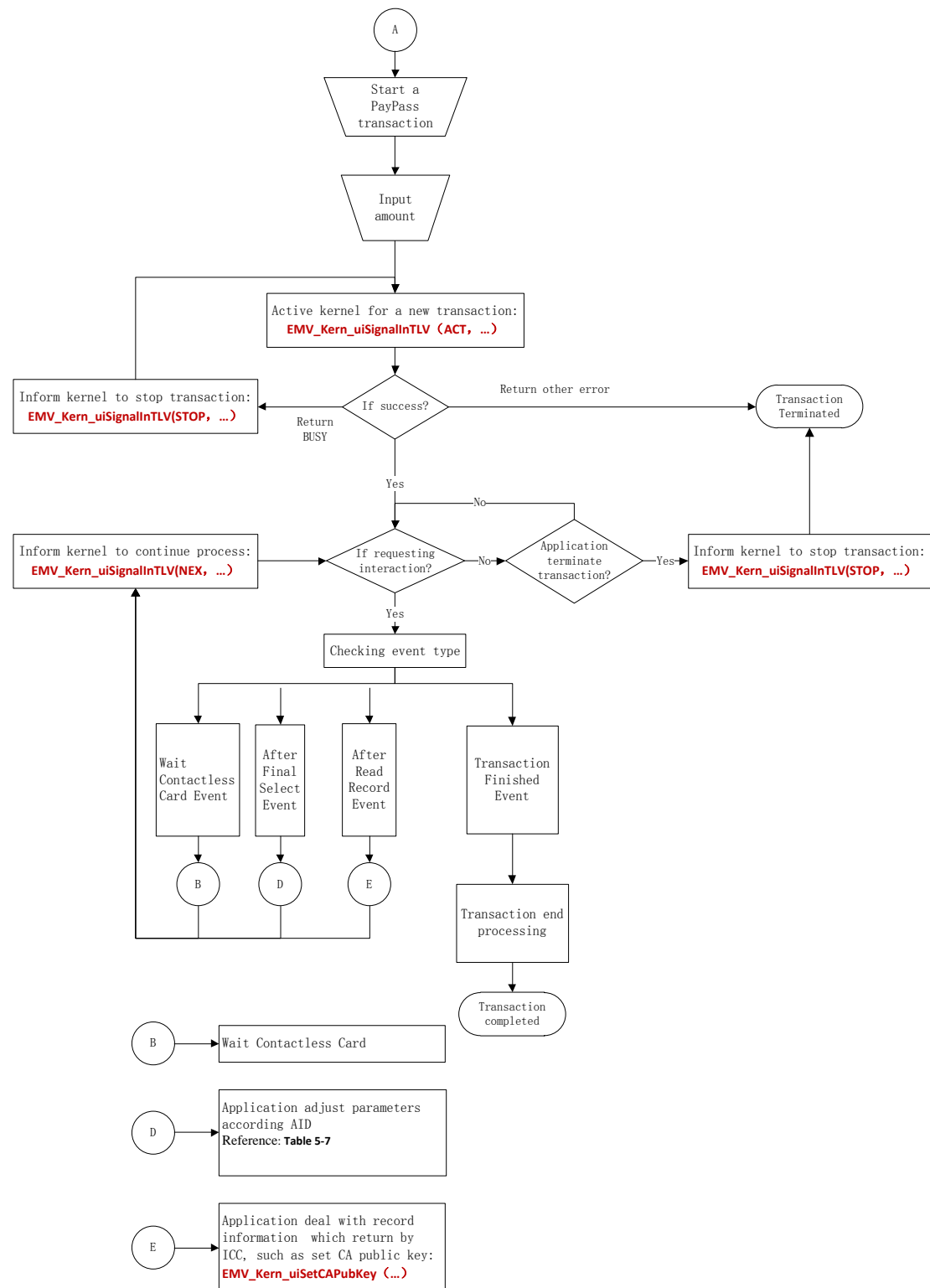




Table 5-7 Transaction Parameter requirement for PayPass

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement			
<b>D: After Final Selection</b>	After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for paypass flow:			
	Parameters Tag	property	Reference value	Default value
	EMV_TAG_TM_TERMTYPE	M	0x22	
	EMV_TAG_TM_TRANSTYPE	O	0x00(Purchase)	0x00
	EMV_TAG_TM_CAP	O	0xE0F808	0x000000
	EMV_TAG_TM_CAP_AD	O	0x6000F0A001	0x0000000000 0
	EMV_TAG_TM_TRANSDATE	M	0x171216	
	EMV_TAG_TM_TRANSTIME	M	0x131535	
	EMV_TAG_TM_AUTHAMNTN	M	0x000000000100	
	EMV_TAG_TM_CNTRYCODE	M	0x0156	
	EMV_TAG_TM_CURCODE	M	0x0156	
	DEF_TAG_TAC_DECLINE	O	0x840000000C	0x8400000000 C
	DEF_TAG_TAC_ONLINE	O	0x840000000C	0x8400000000 C
	DEF_TAG_TAC_DEFAULT	O	0x840000000C	0x8400000000 C
	DEF_TAG_M_TRANS_MOD	O	0x02	0x02
	DEF_TAG_M_BALANCE_SUP	O	0x00	0x00
	DEF_TAG_M_TORN_TRANS	O	0x00	0x00
	DEF_TAG_M_CDV_SUP	O	0x00	0x00
	M_TAG_TM_TRANS_LIMIT	M	0x000000030000 (300.00)	0x0000000000 000
	M_TAG_TM_TRANS_LIMIT_CDV	M	0x000000050000 (500.00)	0x0000000000 000
	M_TAG_TM_CVM_LIMIT	O	0x000000001000( 10.00)	0x0000000000 000
	M_TAG_TM_FLOOR_LIMIT	O	0x000000001000( 100.00)	0x0000000000 000
	DEF_TAG_M_REQ_CVM	O	0x60	0x00
	DEF_TAG_M_REQ_NOCVM	O	0x08	0x00
	DEF_TAG_M_MAG_REQ_CVM	O	0x10	0xF0

	DEF_TAG_M_MAG_REQ_NOCVM	O	0x00	0xF0
	EMV_TAG_TM_CUREXP	O	0x02	
	M_TAG_TM_9F7C	O	0x0101010101010101 0101010101010101 0101010101010101	
	M_TAG_TM_9F53	O	0x01	
	M_TAG_TM_9F6D	O	0x0001	0x0001
	<p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p>EMV_Kern_uiSetTLV</p> <p>EMV_Kern_uiSetTLVList</p>			

## 5.8 AMEX

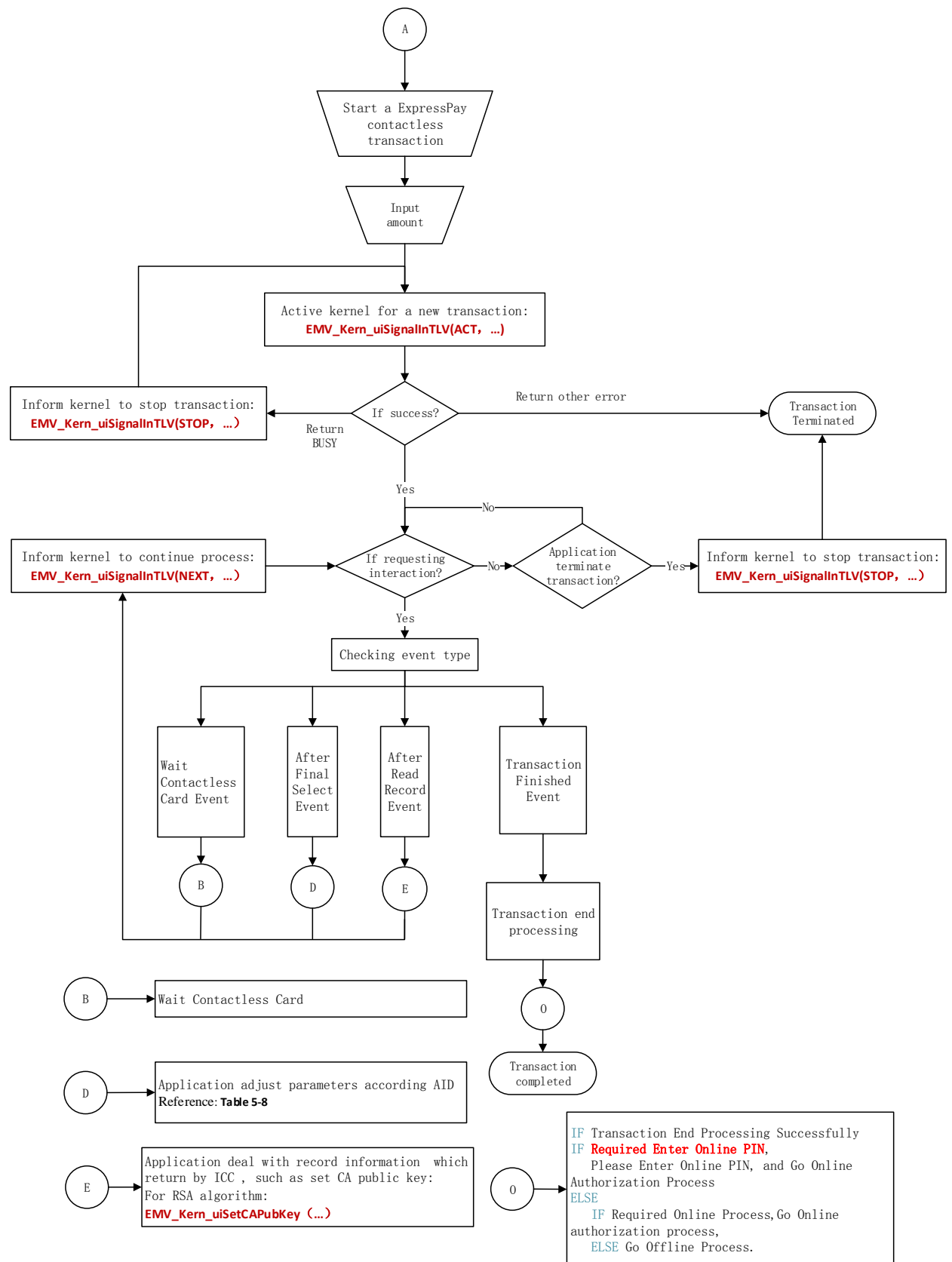


Table 5-8 Transaction Parameter requirement for AMEX

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>ACT: active kernel of a new transaction</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03</li> <li>DEF_TAG_PPSE_6A82_TURNTO_AIDLIST(M) REF_V:0x01</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p>
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. All parameters Tag are required to set at this step as bellow:</p> <ul style="list-style-type: none"> <li>EMV_TAG_TM_TERMTYPE(M) REF_V:0x22</li> <li>EMV_TAG_TM_CAP(M) REF_V:0xE0E8C8</li> <li>EMV_TAG_TM_CAP_AD(M) REF_V:0x6000F0B001</li> <li>A_TAG_TM_9F6D(M) REF_V:0xC0</li> <li>A_TAG_TM_9F6E(M) REF_V:0xD8E00000</li> <li>EMV_TAG_TM_CNTRYCODE(M) REF_V:0x0620</li> <li>EMV_TAG_TM_CURCODE(M) REF_V:0x0978</li> <li>EMV_TAG_TM_APPVERNO(M) REF_V:0x0001(Fixed Value)</li> <li>DEF_TAG_TAC_DECLINE(O) REF_V:0x0000000000</li> <li>DEF_TAG_TAC_ONLINE(O) REF_V:0x0000000000</li> <li>DEF_TAG_TAC_DEFAULT(O) REF_V:0x0000000000</li> <li>A_TAG_TM_TRANS_LIMIT(O) REF_V:0x000000015000</li> <li>A_TAG_TM_FLOOR_LIMIT(O) REF_V:0x000000010000</li> <li>A_TAG_TM_CVM_LIMIT(O) REF_V:0x000000005000</li> <li>EMV_TAG_TM_FLOORLMT(O)<sup>2</sup> REF_V:0x00002710(100.00)</li> <li>EMV_TAG_TM_TRANSTYPE(M) REF_V:0x00(Purchase)</li> <li>EMV_TAG_TM_AUTHAMNTN(M) REF_V:0x00000000100(1.00)</li> <li>EMV_TAG_TM_OTHERAMNTN(O) REF_V:0x0000000000</li> <li>EMV_TAG_TM_TRANSDATE(M) REF_V:0x161216</li> <li>EMV_TAG_TM_TRANSTIME(M) REF_V:0x161535</li> <li>EMV_TAG_TM_TRSEQCNTR(O) REF_V:0x00000001</li> <li>A_TAG_PREAGAIN(C) REF_V:0x00 or 0x01</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p>
<b>E: After Read Application Data</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> </ul>

<sup>2</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B) is 4 Bytes Hex Code, Such as \$100.00, Setting need like this “\x00\x00\x27\x10”.

	<ul style="list-style-type: none"> <li>• A_TAG_TM_TRANS_LIMIT(C) REF_V:Sets of Dynamic Reader Limits</li> <li>• A_TAG_TM_FLOOR_LIMIT(C) REF_V:Sets of Dynamic Reader Limits</li> <li>• A_TAG_TM_CVM_LIMIT(C) REF_V:Sets of Dynamic Reader Limits</li> <li>• A_TAG_TM_IN_CARD_BIN_RANGE(O) REF_V:0xA0 or 0x00</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p>
--	--

Remarks:

The EMVKernel will not required an 'Online Authorization Event' when the transaction performs an Approved offline. EMVKernel will only required ONLINE\_PIN at 'Transaction End Event', and Application can gets CVM Type from data of 'EMV\_tTransData->ucCVM' and perform CVM indicate the cardholder.

*The following underlined content is only using for Kernel Version which is less than or equal to Ver17-11-08 Verison.*

*About Return Code, you can refer the table bellow priority, and then refer the [Annex C Transaction Return Code](#).*

```
#define AMEX_ACTION_SWITCH_INTERFACE 0xF40000E1//18.2.1 Try Another Interface
#define AMEX_ACTION_REQUEST_ANOTHER_PAYMENT 0xF40000E2//18.2.3 Try Another Payment
#define AMEX_ACTION_RESTART_TRY_AGAIN 0xF40000E3//Start,And Try Again
#define AMEX_ACTION_ENDAPPLICATION 0xF40000E5//End Application
```

*The Return Code about 'AMEX\_ACTION\_RESTART\_TRY\_AGAIN' which return from kernel when do an Visa transaction commented as bellow:*

*In 2 Steps (which Mobile CVM performed but failed in CVM Step or which Card return 6984 in TAA1 step), Kernel will return AMEX\_ACTION\_RESTART\_TRY\_AGAIN, and application need power off RF field period 1 to 3 seconds (default 1.5s) and power on it again, and prompt the cardholder the message "See Phone for Instructions", application shall indicates A\_TAG\_PREAGAIN(Tag: DF8130) to 0x01 to the Kernel, and then present the card again do the transaction to completion. During the second Present Card, the context of the current transation persists unchanged, including Amount, Authorized and Transaction Currency code, and Pre-Condition and Pre-Preprocessing data remains unchanged.*

## 5.9 DISCOVER

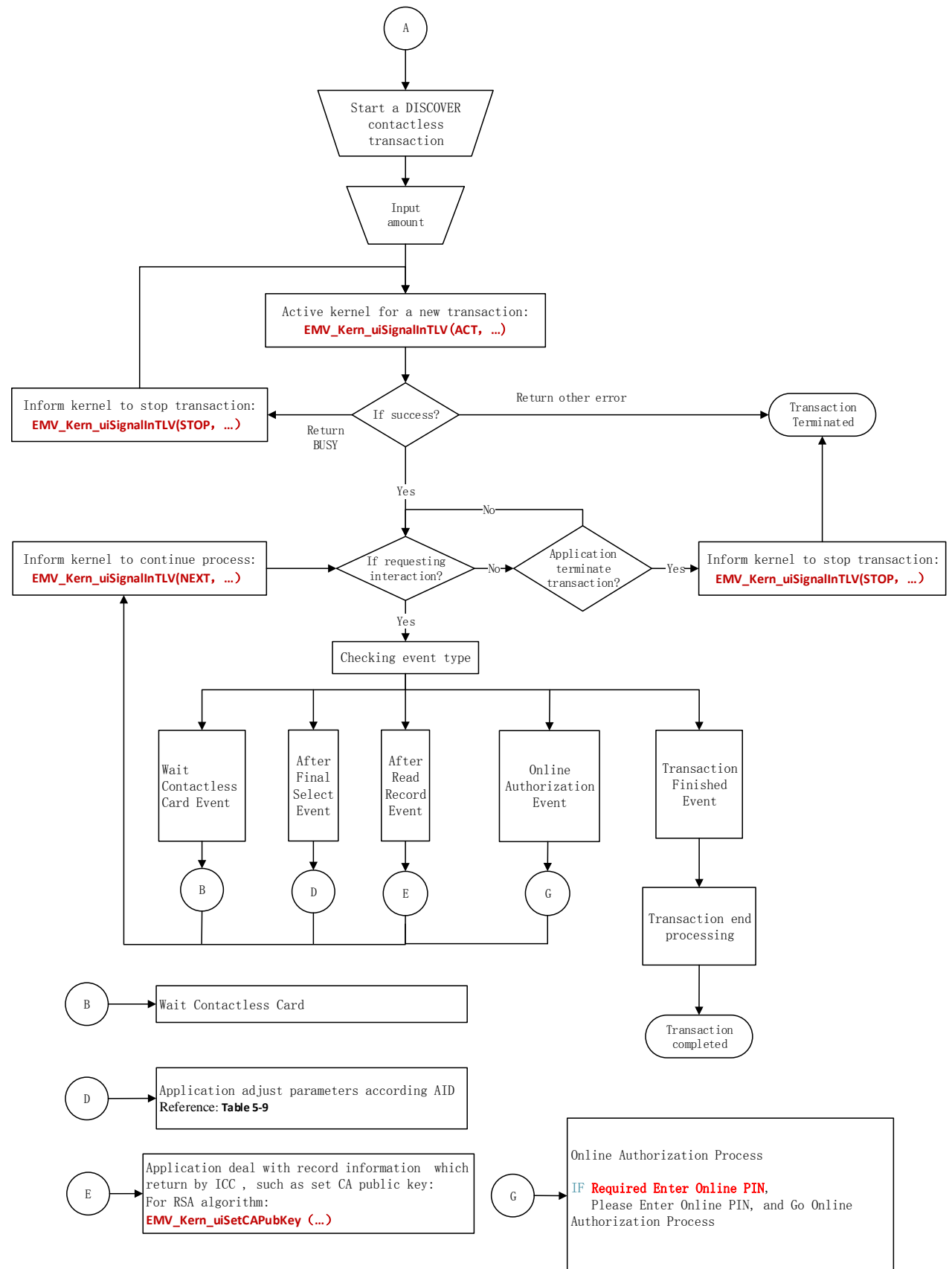


Table 5-9 Transaction Parameter requirement for DISCOVER

REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>ACT: active kernel of a new transaction</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03(D-PAS), 0x04(ZIP)</li> <li>EMV_TAG_TM_AUTHAMNTN(M) REF_V:0x00000000100(1.00)</li> <li>EMV_TAG_TM_OTHERAMNTN(O) REF_V:0x00000000000</li> <li>D_TAG_TM_9F66(M) REF_V:0x26004000</li> <li>D_TAG_TM_RD_RCP(O) REF_V:0xFC80</li> <li>D_TAG_TM_TRANS_LIMIT(O) REF_V:0x000000015000</li> <li>D_TAG_TM_FLOOR_LIMIT(O) REF_V:0x000000010000</li> <li>D_TAG_TM_CVM_LIMIT(O) REF_V:0x000000005000</li> <li>EMV_TAG_TM_FLOORLMT(O)<sup>3</sup> REF_V:0x00002710(100.00)</li> <li>DEF_TAG_D_ISSUERSCRIPT_EXCUTIVE(C) REF_V:0x01 or 0x00</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. All parameters Tag are required to set at this step as bellow:</p> <ul style="list-style-type: none"> <li>EMV_TAG_TM_TERMTYPE(M) REF_V:0x22</li> <li>EMV_TAG_TM_CAP(M) REF_V:0xE0E8C8</li> <li>EMV_TAG_TM_CAP_AD(M) REF_V:0x6000F0B001</li> <li>EMV_TAG_TM_CNTRYCODE(M) REF_V:0x0840</li> <li>EMV_TAG_TM_CURCODE(M) REF_V:0x0840</li> <li>EMV_TAG_TM_APPVERNO(M) REF_V:0x0100</li> <li>DEF_TAG_TAC_DECLINE(O) REF_V:0x00000000000</li> <li>DEF_TAG_TAC_ONLINE(O) REF_V:0x00000000000</li> <li>DEF_TAG_TAC_DEFAULT(O) REF_V:0x00000000000</li> <li>EMV_TAG_TM_TRANSTYPE(M) REF_V:0x00(Purchase)</li> <li>EMV_TAG_TM_TRANSDATE(M) REF_V:0x161216</li> <li>EMV_TAG_TM_TRANSTIME(M) REF_V:0x161535</li> <li>EMV_TAG_TM_TRSEQCNTR(O) REF_V:0x000000001</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>
<b>E: After Read Application Data</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> </ul>

<sup>3</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B) is 4 Bytes Hex Code, Such as \$100.00, Setting need like this “\x00\x00\x27\x10”.

	<p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>
<p><b>G:After Online Authorization Process</b></p>	<ul style="list-style-type: none"> <li>• DEF_TAG_ONLINE_STATUS(C) REF_V:0x00 or 0x01</li> <li>• EMV_TAG_TM_ARC(C) REF_V:ARC from Host</li> <li>• DEF_TAG_AUTHORIZE_FLAG(C) REF_V:0x00 or 0x01</li> <li>• EMV_TAG_TM_AUTHCODE(C) REF_V:Var.</li> <li>• DEF_TAG_HOST_TLVDATA(C) REF_V:Var.,such as Issuer Script</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p> <p><b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>

Remarks:

The EMVKernel will not required an 'Online Authorization Event' when the transaction performs an Approved offline. EMVKernel will only required ONLINE\_PIN at 'Transaction End Event', and Application can gets CVM Type from data of 'EMV\_tTransData->ucCVM' and perform CVM indicate the cardholder.

But, the EMVKernel will required an 'Online Authorization Event' when the transaction performs an Online Required, and Application shall check it if the EMVkernel required an CVM which can gets CVM Type from data of 'EMV\_tTransData->ucCVM' and perform CVM indicate the cardholder, and continue the transaction to Acquirer,and to the End Event. The EMVKernel only required a CVM a time throughout the transaction.

*The following underlined content is only using for Kernel Version which is less than or equal to Ver17-11-08 Verison.*

*About Return Code, you can refer the table bellow priority, and then refer the [<Annex C Transaction Return Code>](#).*

```
#define DISCOVER_ACTION_SWITCH_INTERFACE 0xF60000E1//Try Another Interface
#define DISCOVER_ACTION_REQUEST_ANOTHER_PAYMENT 0xF60000E2//Try Another Payment
#define DISCOVER_ACTION_RESTART_TRY_AGAIN 0xF60000E3//Start,And Try Again
#define DISCOVER_ACTION_ENDAPPLICATION 0xF60000E5//End Application(Terminate)
#define DISCOVER_ACTION_USE_ANOTHER_CARD 0xF60000E7//Try Another Card(Ask user to use another card)
```

*The Return Code about 'DISCOVER\_ACTION\_RESTART\_TRY\_AGAIN' which return from kernel when do an Visa transaction commented as bellow:*

*In GPO Step, Kernel will return DISCOVER\_ACTION\_RESTART\_TRY\_AGAIN when card return SW=6986, and application need not power off RF field, and prompt the cardholder the message "Please Removed Card, Present Card Again", then present the card again do the transaction to completion. During the second Present Card, the context of the current transation persists*



*unchanged, including Amount, Authorized and Transaction Currency code, and Pre-Condition and Pre-Preprocessing data remains unchanged.*

## 5.10 JCB

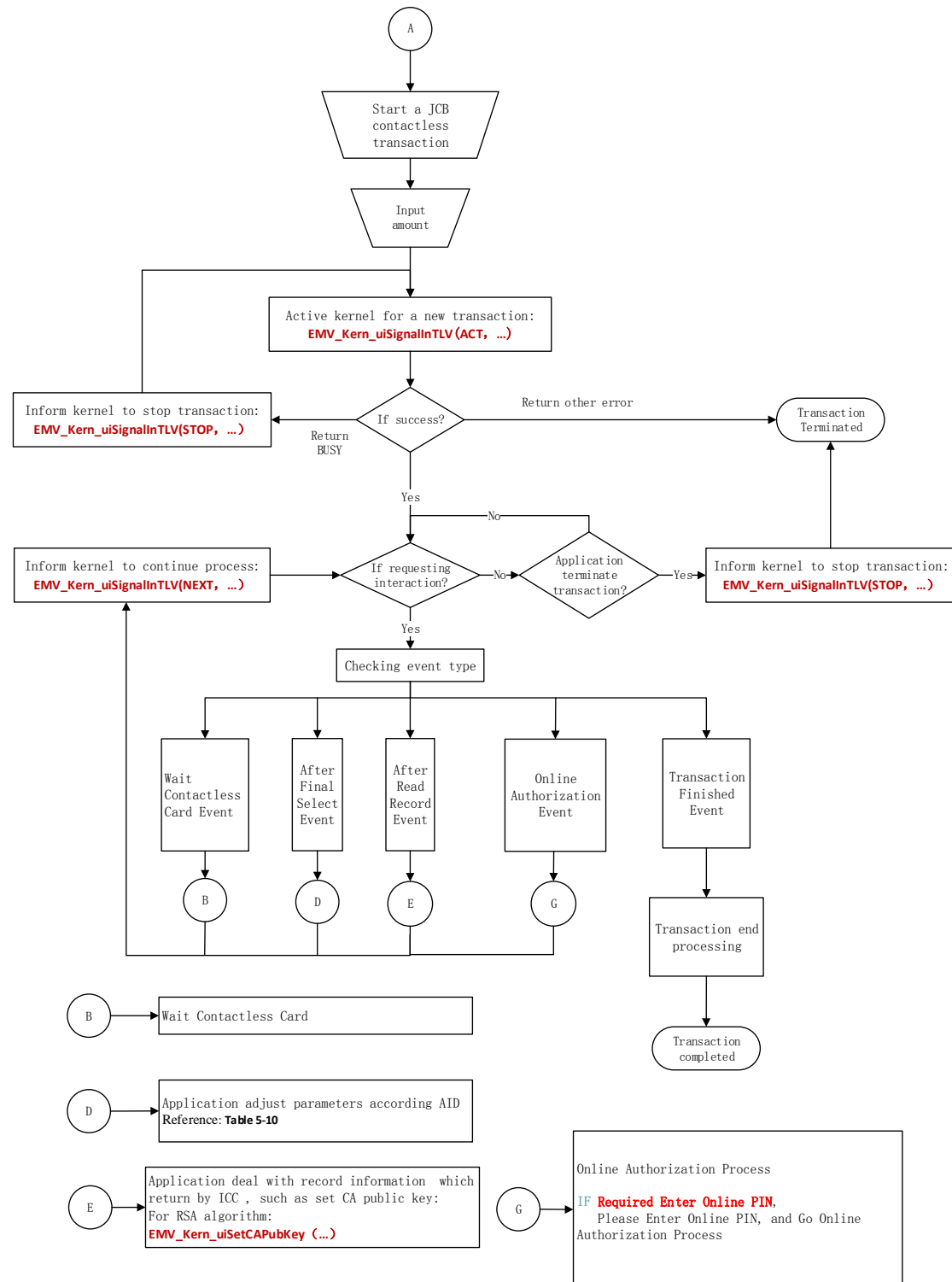
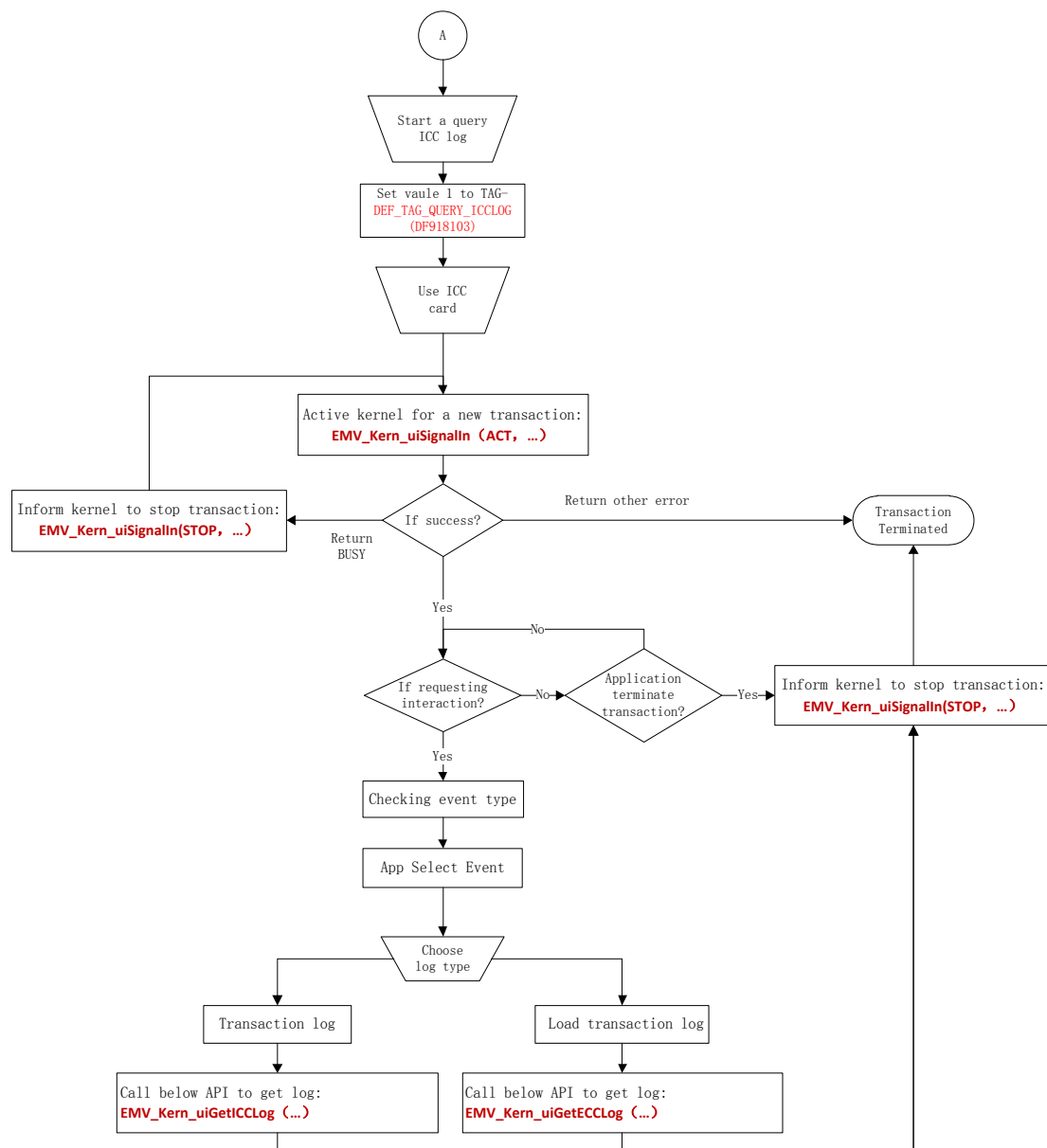


Table 5-10 Transaction Parameter requirement for JCB

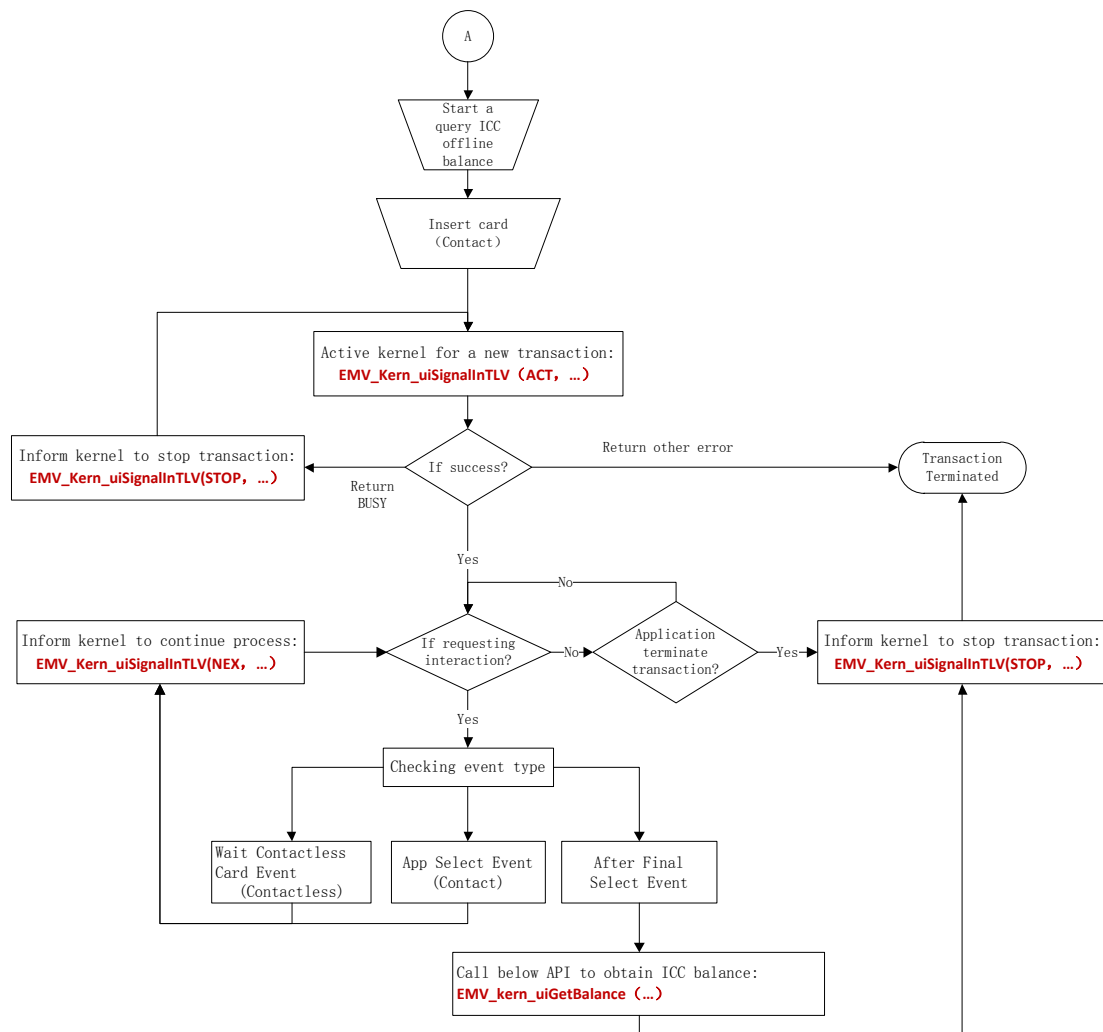
REF\_V-Reference Value M:Mandatory O-Optional C-Conditional

Event Type	Transaction Parameters Requirement
<b>D: After Final Selection</b>	<p>After final application selection, the AID is decided. Some parameters can be set with different value base on different AID. The following parameters are recommended for paypass flow:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN(M) REF_V(6):0x00000000100(1.00)</li> <li>• EMV_TAG_TM_TRANSTYPE(M) REF_V(1):0x00(Purchase)</li> <li>• EMV_TAG_TM_TERMTYPE(M) REF_V(1):0x22</li> <li>• DEF_TAG_J_COMB_OPTION(M) REF_V(2):0x7B00</li> <li>• DEF_TAG_J_TIP(M) REF_V(3):0x708000</li> <li>• EMV_TAG_TM_CNTRYCODE(M) REF_V(2):0x0392</li> <li>• EMV_TAG_TM_CURCODE(M) REF_V(2):0x0392</li> <li>• EMV_TAG_TM_CUREXP(M) REF_V(1):0x02</li> <li>• EMV_TAG_TM_TRANSDATE(M) REF_V:0x161216</li> <li>• EMV_TAG_TM_TRANSTIME(M) REF_V:0x161535</li> <li>• EMV_TAG_TM_OTHERAMNTN(O) REF_V(6):0x0000000000</li> <li>• DEF_TAG_TAC_DECLINE(O) REF_V(5):0x0410000000</li> <li>• DEF_TAG_TAC_ONLINE(O) REF_V(5):0x9060009000</li> <li>• DEF_TAG_TAC_DEFAULT(O) REF_V(5):0x9040008000</li> <li>• EMV_TAG_TM_ACQID(O) REF_V(6):0x000000000010</li> <li>• EMV_TAG_TM_MCHCATCODE(O) REF_V(2):0x7032</li> <li>• EMV_TAG_TM_MCHNAMELOC(O) REF_V(0_32):0x5858204D45524348414E54205959204C4F434154494F4E(23byte)</li> <li>• DEF_TAG_J_TRANS_LIMIT(O) REF_V(6):0x000000020000(200.00)</li> <li>• DEF_TAG_J_CVM_LIMIT(O) REF_V(6):0x000000010000(100.00)</li> <li>• DEF_TAG_J_FLOOR_LIMIT(O) REF_V(6):0x000000004500(45.00)</li> <li>• DEF_TAG_J_RS_MAX_PERCENT(O) REF_V(1):0x00</li> <li>• DEF_TAG_J_RS_TARGET_PERCENT(O) REF_V(1):0x00</li> <li>• DEF_TAG_J_RS_THRESH_VALUE(O) REF_V(4):0x000007D0(20.00)</li> </ul> <p>Tag Definition refer to <a href="#">Annex A</a></p> <p>Via API:</p> <p><b>EMV_Kern_uiSetTLV</b></p> <p><b>EMV_Kern_uiSetTLVList</b></p>

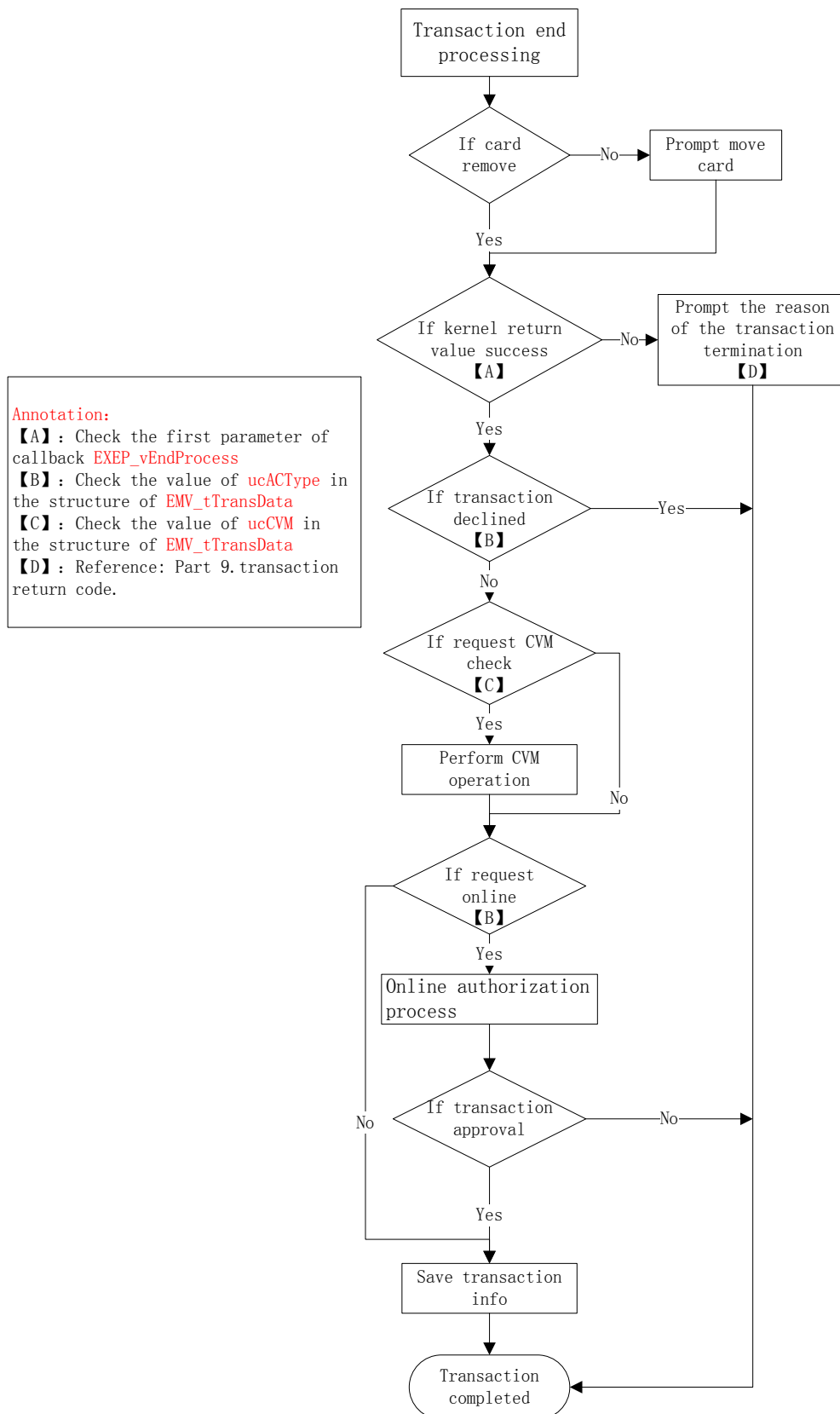
## 5.11 Query ICC Transaction log



## 5.12 Query ICC Offline Balance



## 5.13 Transaction Finish Process



## 6. API Specification

### 6.1 EMV\_Kern\_uiCreateObject

<b>Prototype</b>	uint EMV_Kern_uiCreateObject(const EMV_Configuration *ptEMVConfiguration, EMV_OBJECT *ptEMVObject)		
<b>Function</b>	Create an EMV kernel Application Object.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
ptEMVConfiguration	<b>In</b>	Configuration	Refer to: <a href="#">EMV Configuration</a>
ptEMVObject	<b>Out</b>	An Application Object pointer	
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS		
other	CreateObject was failed		

#### Program Guide:

This API should be executed before any other APIs defined in this document. After this API executing, EMV Kernel will be turned into initialization mode. Any parameter which had been stored in EMV Kernel will be cleared.

## 6.2EMV\_Kern\_uiDestroyObject

<b>Prototype</b>	uint EMV_Kern_uiDestroyObject(EMV_OBJECT tEMVObject)		
<b>Function</b>	Destroy an Application Object which was create suces before.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	After call this function, Application need put this parameter to NULL.
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS		

### Program Guide:

Destroy a previously created EMV application object.

### 6.3EMV\_Kern\_uiManageAID

<b>Prototype</b>	uint EMV_Kern_uiManageAID(EMV_OBJECT tEMVObject, uchar ucAction, uchar ucAIDLen, const uchar *pauAID, uchar ucPartSlt, uchar ucKID)		
<b>Function</b>	AID Management		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ucAction	<b>In</b>	EMV_FLAG_ADD-ADD AID EMV_FLAG_DELETE-Delete AID EMV_FLAG_CLEAR-Clear AID	Action Flag. Refer to: <a href="#">Macro Definition</a>
ucAIDLen	<b>In</b>	5-16	AID length
pauAID	<b>In</b>	Hex	AID
ucPartSlt	<b>In</b>	0 or 1	Part ial matching: 0-don't support 1-support
ucKID	<b>In</b>	Kernel ID, Support list: EMV_KERNELID_EMV -0x00 EMV_KERNELID_MASTER -0x02 EMV_KERNELID_VISA -0x03 EMV_KERNELID_AMEX -0x04 EMV_KERNELID_JCB -0x05 EMV_KERNELID_DISCOVER -0x06 EMV_KERNELID_PBOC -0x07 EMV_KERNELID_NSICC -0xDA	0- mean undefined Kernel ID
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0xF1	AID length error.		
0xF2	AID Pointer is NULL.		
0xF3	Action Flag is illegal.		
0xF4	AID list space is FULL(Max of AID number is <= 100).		
0xF5	Partial matching flag is illegal.		
Other Value	Other Error.		

#### Program Guide:

Before any transaction activating, application should add one or more AID. Only when AID is supported by both application and IC card, will AID be shown in candidate list.



## 6.4EMV\_Kern\_uiUpdateCAIndexList

<b>Prototype</b>	uint EMV_Kern_uiUpdateCAIndexList(EMV_OBJECT tEMVObject, const uchar *pauRID, const uchar *pauIndexList, uchar ucListLen)		
<b>Function</b>	Updating CA public key		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
pauRID	In	Hex	RID
pauIndexList	In	Hex	CA public key index list
ucListLen	In	0-50	Length of index list. If zero, it will clear index list which had been stored in kernel.
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x01	(Max of CA public key index is <= 6)FULL.		
0xF1	RID is NULL.		
0xF2	Value of list length exceeds the max value.		
Other Value	Other Error.		

### Program Guide:

The API is designed for qPBOC and qVSDC. It will reduce the risk of transaction offline failure owing to CA public key absence which will cause IC card offline balance decrease.

## 6.5 EMV\_Kern\_uiSetCAPubKey

<b>Prototype</b>	uint EMV_Kern_uiSetCAPubKey(EMV_OBJECT tEMVObject, const EMV_tPKFILESTRU *ptPubKey)		
<b>Function</b>	CA Public key setting for RSA		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ptPubKey	<b>In</b>	Structure EMV_tPKFILESTRU pointer or NULL	Refer to: <a href="#">EMV_tPKFILESTRU</a> If NULL, Clear stored in kernel.
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x01	Checking Hash value fail.		
0xF2	Format error.		
Other Value	Other Error.		

### Program Guide:

The API should be executed when “after read record event” occurs. Application should set CA public key according to RID and key index which are responded by “after read record event”. EMV kernel will keep only one group of public key data which is set the latest.

## 6.6 EMV\_Kern\_uiSetCAPubKey\_SM

<b>Prototype</b>	uint EMV_Kern_uiSetCAPubKey_SM(EMV_OBJECT tEMVObject, const EMV_tPKFILESTRU_SM *ptPubKey)		
<b>Function</b>	CA Public key setting for SM		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
ptPubKey	In	Structure EMV_tPKFILESTRU_SM pointer or NULL	Refer to: <a href="#">EMV tPKFILESTRU_SM</a> If NULL, Clear stored in kernel.
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0xF2	Format error.		
Other Value	Other Error.		

### Program Guide:

The API should be executed when “after read record event” occurs. Application should set CA public key according to RID and key index which are responded by “after read record event”. EMV kernel will keep only one group of public key data which is set the latest.

## 6.7 EMV\_Kern\_uiManageRecCert

<b>Prototype</b>	uint EMV_Kern_uiManageRecCert(EMV_OBJECT tEMVObject, uchar ucAction, const EMV_tRecCert *ptRecCert)		
<b>Function</b>	Revocate Public Key Certificate Management		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ucAction	<b>In</b>	EMV_FLAG_ADD-Add Key EMV_FLAG_DELETE -Delete Key EMV_FLAG_CLEAR -Clear Key	Action Flag. Refer to <a href="#">Macro Definition</a>
ptRecCert	<b>In</b>	Structure EMV_tRecCert pointer	Refer to: <a href="#">EMV tRecCert</a>
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x10	FULL.		
0x11	Had existed, reset it.		
0xE1	Action Flag illegal.		
0xE2	Parameter error.		
Other Value	Other Error.		

### Program Guide:

This interface is used to manage the list of CA public key recovery certificates maintained by the EMV kernel, which can store up to 100.

## 6.8 EMV\_Kern\_uiManageDOL

<b>Prototype</b>	uint EMV_Kern_uiManageDOL(EMV_OBJECT tEMVObject, uchar ucFlag, uchar ucActon, uchar *paDoL, uchar *pcLen)		
<b>Function</b>	DDOL、TDOL、UDOL Management.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ucFlag	<b>In</b>	1-DDOL 2-TDOL 3-UDOL	DOL Flag
ucAction	<b>In</b>	0x01-Set DOL 0x02-Get DOL	Action Flag
paDoL	<b>In\Out</b>	Hex	If Set DOL, it is anew DOL If Get DOL, it will be return DOL.
pcLen	<b>In\Out</b>	Length pointer	If Set DOL, it is length of new DOL If Get DOL, it is length of return DOL
<b>Return</b>	<b>Meaning</b>		
0	Success.		
0x01	Parameter fail.		
0x02	Action Flag illegal.		
0x03	Dol Flag illegal.		
0x11/0x21/0x31	DOL length error.		
0x12/0x22/0x32	DOL Format error.		
Other Value	Other Error.		

### Program Guide:

For example:

#### Setting DDOL="9F3704":

```
Uchar ucLen;
```

```
ucLen=3;
```

```
uiRet = EMV_Kern_uiManageDOL(1, 1, (uchar*)"9F3704", &ucLen);
```

#### Get DDOL:

```
Uchar ucLen;
```

```
Uchar auBuffer;
```

```
ucLen=3;
```

```
uiRet = EMV_Kern_uiManageDOL(1, 2, auBuffer, &ucLen);
```

## 6.9 EMV\_Kern\_uiSignalInTLV

<b>Prototype</b>	uint EMV_Kern_uiSignalInTLV(EMV_OBJECT tEMVObject, uchar ucSignal, const uchar *pauTLVData, uint uiTLVDataLen)		
<b>Function</b>	Transaction signal entry		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
ucSignal	In	EMV_SIGNAL_ACT EMV_SIGNAL_NEXT EMV_SIGNAL_STOP EMV_SIGNAL_CLEAN	Signal Type , refer to: <a href="#">Macro Definition</a>
pauTLVData	In	NULL or Signal parameter pointer	TLV Format parameters for activate a new transaction or response for different event requesting during transaction, reference <a href="#">Table 4-1</a> .
uiTLVDataLen	In	0(If signal parameter is NULL) Length of signal parameter	Length of TLV parameters
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
EMV_RESULT_BUSY	EMV thread is busy, try again later.		
0xF001	Signal can't be accepted currently.		
0xF0F1	Parameters don't format with TLV		
0xF1XX	Signal type is illegal.		
0xF201	Length of parameter error.		
Other Value	Other Error.		

### Program Guide:

// Activate a new transaction with only PSE way.

```
uchar auTLVData[100];
```

```
uint uiTLVLen;
```

```
uint uiRet;
```

```
uiTLVLen=0;
```

```
memset(auTLVData, 0, sizeof(auTLVData));
```

```
memcpy(auTLVData+uiTLVLen, DEF_TAG_PSE_FLAG, sizeof(DEF_TAG_PSE_FLAG));
```

```
uiTLVLen+= sizeof(DEF_TAG_PSE_FLAG);
```

```
auTLVData[uiTLVLen++]=0x01;
```

```
auTLVData[uiTLVLen++]=0x01;
```

```
uiRet =EMV_Kern_uiSignalInTLV(EMV_SIGNAL_ACT, auTLVData, uiTLVLen);
```

```
if(uiRet !=0) return fail.
```



## 6.10 EMV\_Kern\_uiSetTLV

<b>Prototype</b>	uint EMV_Kern_uiSetTLV(EMV_OBJECT tEMVObject, uchar ucKernelID, const uchar *paTag, uchar ucValueLen, const uchar *paValue)		
<b>Function</b>	Set TLV data element.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
ucKernelID	In	Hex	Tag name defined with a prefix such as: EMV_TAG_XXXX (For EMV) C_TAG_XXXX (For PBOC ) V_TAG_XXXX (For VISA) M_TAG_XXXX (For MASTER) A_TAG_XXXX (For AMEX) D_TAG_XXXX (For DISCOVER) J_TAG_XXXX (For JCB) DEF_TAG_XXXX (For DEFINE) Reference <a href="#">Annex A TAG Dictionary</a> .
paTag	In	Hex	Tag
ucValueLen	In	0 or 1~0xFF	Value length If zero, clear stored one in kernel
paValue	In	NULL(only for length is 0) Hex	Value or NULL
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x00F0	Parameter error		
0x00F1	ICC Element, can't be set.		
0x00F2	Kernel ID is illegal		
0x00E1	Format error.		
Other Value	Other Error.		

### Program Guide:

For example:

POS Enter mode setting:

```
uiRet = EMV_Kern_uiSetTLV(EMV_KERNELID_EMV, (uchar*)"x9F\x39", 1, (uchar*)"x91");
```

Master Card Mobile Support Indicator setting:

```
uiRet = EMV_Kern_uiSetTLV(EMV_KERNELID_MASTER, (uchar*)"x9F\x7E", 1, (uchar*)"x02");
```

Clear terminal serial number stored in kernel:

```
uiRet = EMV_Kern_uiSetTLV(EMV_KERNELID_EMV, (uchar*)"x9F\x1E", 0, NULL);
```



## 6.11 EMV\_Kern\_uiSetTLVList

Prototype	uint EMV_Kern_uiSetTLVList(EMV_OBJECT tEMVObject, uchar ucKernelID, const uchar *pauData, uint uiDataLen)		
Function	TLV list setting		
Parameter	Type	Value Range	Specification
tEMVObject	In	A point to an Application Object	
ucKernelID	In	Hex	<p>Tag name defined with a prefix such as:</p> <p>EMV_TAG_XXXX (For EMV)</p> <p>C_TAG_XXXX (For PBOC )</p> <p>V_TAG_XXXX (For VISA)</p> <p>M_TAG_XXXX (For MASTER)</p> <p>A_TAG_XXXX (For AMEX)</p> <p>D_TAG_XXXX (For DISCOVER)</p> <p>J_TAG_XXXX (For JCB)</p> <p>DEF_TAG_XXXX (For DEFINE)</p> <p>Reference <a href="#">Annex A TAG Dictionary</a>.</p> <p><b>Note:</b></p> <p>Tags defined by PBOC or VISA or MASTER, can not be set into kernel at the same time.</p> <ul style="list-style-type: none"> <li>• If TLV list including PBOC Tags, Kernel ID should be EMV_KERNELID_PBOC</li> <li>• If TLV list including VISA Tags, Kernel ID should be EMV_KERNELID_VISA</li> <li>• If TLV list including MASTER Tags, Kernel ID should be EMV_KERNELID_MASTER</li> <li>• If TLV list including AMEX Tags, Kernel ID should be EMV_KERNELID_AMEX</li> <li>• If TLV list including DISCOVER Tags, Kernel ID should be EMV_KERNELID_DISCOVER</li> <li>• If TLV list including JCB Tags, Kernel ID should be EMV_KERNELID_JCB</li> <li>• If TLV list excluding PBOC and VISA and MASTER Tag, Kernel ID</li> </ul>

			should be EMV_KERNELID_EMV or EMV_KERNELID_DEFINE
pauData	In	Hex	TLV list
uiDataLen	In	>=1	TLV list length
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
Other Value	Other Error.		

#### Program Guide:

For example:

**State 1:** Only EMV Tags, such as: **EMV\_TAG\_TM\_CNTRYCODE**, **EMV\_TAG\_TM\_CURCODE**

auTLVlist= “\x9F\x1A\x02\x08\x40\x5F\x2A\x02\x01\x56”

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_EMV**, auTLVlist, 10);

**State 2:** Only DEFINE Tags, such as: **DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

auTLVlist= “\xDF\x91\x81\x01\x01\x00\xDF\x91\x81\x03\x01\x01”

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_DEFINE**, auTLVlist, 12);

**State 3:** Including EMV and/ or DEFINE Tags, such as:**EMV\_TAG\_TM\_CNTRYCODE, EMV\_TAG\_TM\_CURCODE, DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_EMV**,……);

**State 4:** Including PBOC and EMV and /or DEFINE Tags, such as:**C\_TAG\_TM\_9F7A, EMV\_TAG\_TM\_CURCODE, DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_PBOC**, ……);

**State 5:** Including VISA and EMV and /or DEFINE Tags, such as:**V\_TAG\_TM\_9F66, EMV\_TAG\_TM\_CURCODE, DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_VISA**, ……);

**State 6:** Including MASTER and EMV and /or DEFINE Tags, such as:**M\_TAG\_TM\_TRANS\_LIMIT, EMV\_TAG\_TM\_CURCODE, DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

ucRet = EMV\_Kern\_uiSetTLV(**EMV\_KERNELID\_MASTER**,……);

## 6.12 EMV\_Kern\_uiGetTLV

<b>Prototype</b>	uint EMV_Kern_uiGetTLV(EMV_OBJECT tEMVObject, const uchar *pTag, uchar ucTagLen, uchar *pVal, uint *puiLen)		
<b>Function</b>	Obtain the value of TLV element stored in EMV kernel.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
pTag	<b>In</b>	Hex	Tag
ucTagLen	<b>In</b>	1-3	Tag Length
pVal	<b>Out</b>	value pointer	Value of the tag
puiLen	<b>Out</b>	Length pointer	Length of the value
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x01	Not existed		
0xE1	Tag is undefined.		
0xE2	Tag is illegal.		
Other Value	Other Error.		

### Program Guide:

For example:

Get ICC PAN from EMV Kernel:

```
ucRet = EMV_Kern_uiGetTLV((uchar*)"x5A", 1, auValue, &uiLen);
```

### 6.13 EMV\_kern\_uiGetBalance

<b>Prototype</b>	uint EMV_kern_uiGetBalance(EMV_OBJECT tEMVObject, uchar *pauBalance, uchar *pucLen)		
<b>Function</b>	Obtain IC Card offline balance using APDU instruction		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
pauBalance	<b>Out</b>	Hex	Offline Balance
pucLen	<b>Out</b>	Length pointer	Length of Balance
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
Other Value	Other Error.		

#### Program Guide:

This API should be executed after Final Selection event has occurred.

## 6.14 EMV\_kern\_uiGetDataAPDU

<b>Prototype</b>	uint EMV_kern_uiGetDataAPDU(EMV_OBJECT tEMVObject, uchar ucP1, uchar ucP2, uchar *pauValue, ushort *pusValueLen)		
<b>Function</b>	Obtain ICC element using APDU instruction.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
ucP1	In		First Byte of Tag.
ucP2	In		Second Byte of Tag.
pauValue	Out		The value of Tag.
pusValueLen	Out		Length of value.
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
Other Value	Other Error.		

### Program Guide:

This API should be executed after Final Selection event has occurred.

For example:

Obtain ATC from IC Card:

```
uiRet=EMV_kern_uiGetDataAPDU(0x9F, 0x36, &auATC, &usATCLen);
```

## 6.15 EMV\_Kern\_uiGetICCLog

<b>Prototype</b>	uint EMV_Kern_uiGetICCLog(EMV_OBJECT tEMVObject, const EMV_tSelectAID *ptSelectedAID, uchar *pucLogNum, EMV_tICCLog *ptICCLog)		
<b>Function</b>	Obtain ICC transaction log.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ptSelectedAID	<b>In</b>	Structure EMV_tSelectAID Pointer	Refer to: <a href="#">EMV_tSelectAID</a>
pucLogNum	<b>In</b>	0-0xFF	If non zero, means max number of ICC log which application expected.
pucLogNum	<b>Out</b>	0-0xFF	The number of ICC log actually
ptICCLog	<b>Out</b>	Structure EMV_tICCLog list Pointer	Refer to: <a href="#">EMV_tICCLog</a>
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x01	IC card does not support this function.		
Other Value	Other Error.		

### Program Guide:

This API should be executed after Application Selection event has occurred.

Program example:

```
EMV_tICCLog atICCLog[10];
```

```
EMV_tSelectAID tSelectedAID;
```

```
tSelectedAID.ucAIDLen = AIDLen;
```

```
memcpy(tSelectedAID.auAID, AID,AIDLen);
```

```
memset(atICCLog, 0, sizeof(atICCLog));
```

```
ucICCLogNum = 10;//Application expect to obtain 10 logs
```

```
ucRet = EMV_Kern_uiGetICCLog(&tSelectedAID,&ucICCLogNum, atICCLog);
```

## 6.16 EMV\_Kern\_uiGetECCLog

<b>Prototype</b>	uint EMV_Kern_uiGetECCLog(EMV_OBJECT tEMVObject, const EMV_tSelectAID *ptSelectedAID, uchar *pucLogNum, EMV_tECCLog *ptECCLog)		
<b>Function</b>	Obtain ICC load transaction log.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	In	A point to an Application Object	
ptSelectedAID	In	Structure EMV_tSelectAID Pointer	Refer to: <a href="#">EMV_tSelectAID</a>
pucLogNum	In	0-0xFF	If it is non zero, it represents the max number of ICC log that application expects to obtain.
pucLogNum	Out	0-0xFF	The actually obtained number of ICC log.
ptECCLog	Out	Structure EMV_tECCLog list Pointer	Refer to: <a href="#">EMV_tECCLog</a>
<b>Return</b>	<b>Meaning</b>		
0	SUCCESS.		
0x01	IC card does not support this function.		
Other Value	Other Error.		

### Program Guide:

This API should be executed after Application Selection event has occurred.

And, it is only for the IC card which support load transaction; for example, the PBOC3.0 Ecash card.

Program example:

```
EMV_tECCLog atECCLog[10];
EMV_tSelectAID tSelectedAID;
Uchar ucECCLogNum;
```

```
tSelectedAID.ucAIDLen = AIDLen;
memcpy(tSelectedAID.auAID, AID, AIDLen);
```

```
memset(atECCLog, 0, sizeof(atECCLog));
ucECCLogNum = 10; // Application expect to obtain 10 logs
ucRet = EMV_Kern_uiGetECCLog(&tSelectedAID, &ucECCLogNum, atECCLog);
```



### 6.17 EMV\_Kern\_vSwitchDebug

<b>Prototype</b>	void EMV_Kern_vSwitchDebug(EMV_OBJECT tEMVObject, uchar ucMode);		
<b>Function</b>	Deactive or active kernel debug log function.		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ucMode	<b>In</b>	0 - Deactivate 1 - Activate and obtain debug log after transaction finish. 2 - Activate and obtain debug log during the transaction	Refer to: <a href="#">Obtain Kernel Debug Log.</a>
<b>Return</b>	<b>Meaning</b>		
void	SUCCESS.		

**Program Guide:**

Please refer to: [Obtain Kernel Debug Log.](#)

## 6.18 EMV\_kern\_uiSetHandle

<b>Prototype</b>	uint EMV_kern_uiSetHandle(EMV_OBJECT tEMVObject, uchar ucFlag, EMVHandle hHandle)		
<b>Function</b>	Set device handle into EMV kernel		
<b>Parameter</b>	<b>Type</b>	<b>Value Range</b>	<b>Specification</b>
tEMVObject	<b>In</b>	A point to an Application Object	
ucFlag	<b>In</b>	0xA0 - Contact ICC device 0xA1 - Internal contactless ICC device 0xA2 - External contactless ICC device 0xB1 - Pinpad	
hHandle	<b>In</b>	EMVHandle	Device handle
<b>Return</b>	<b>Meaning</b>		
uint	SUCCESS.		
Other value	Illegal flag.		

**Program Guide:**

## 7. Structure Definition

### EMV\_Configuration

Structure Name	Register Configuration	
Type	Parameter Name	Parameter Specification
EMV_EXPAND_BASEFUN	tExAPI_BASE	Refer to: <a href="#">EMV_EXPAND_BASEFUN</a>
EMV_EXPAND_INTERFACE	tExAPI_IFC	Refer to: <a href="#">EMV_EXPAND_INTERFACE</a>
uint	uiConfigDataLen	Length of uiConfigDataLen
uchar	auConfigData[600]	ConfigData
uchar	ucConfigID	ConfigID for EMV
uchar	ucDebugMode	0 - Deactivate 1 - Activate and obtain debug log after transaction finish. 2 - Activate and obtain debug log during the transaction

### EMV\_tPKFILESTRU

Structure Name	RSA CA Public Key	
Type	Parameter Name	Parameter Specification
uchar	auRid[5];	RID
uchar	ucIndex;	Index
uchar	ucModLen;	Length of Mod
uchar	auMod[256];	Mod
uchar	ucExpLen;	Length of exponent:1 or 3
uchar	auExp[3];	Exponent: \x03 or \x01\x00\x01
uchar	auExpDate[4];	Expiry, format as YYYYMMDD, BCD encoded.
uchar	ucHashFig;	Hash verification flag: 1-support 0-nonsupport
uchar	auHash[20];	Hash value.

## EMV\_tPKFILESTRU\_SM

Structure Name	SM CA Public Key	
Type	Parameter Name	Parameter Specification
uchar	<b>auRid[5];</b>	RID
uchar	<b>ucIndex;</b>	Index
uchar	<b>ucKeyLen;</b>	Length of key
uchar	<b>auPubKey[256];</b>	key
uchar	<b>auExpDate[4];</b>	Expiry, format as YYYYMMDD, BCD encoded.
GROUP_PARA	<b>tGroupPara</b>	Always NULL

## EMV\_tSelectAID

Structure Name	AID being selected.	
Type	Parameter Name	Parameter Specification
uchar	ucAIDLen;	AID length
uchar	auAID[EMV_LEN_MAX_AID];	AID Data

## EMV\_tRecCert

Structure Name	Public Key Revocation	
Type	Parameter Name	Parameter Specification
uchar	ucIndex;	CA public key index.
uchar	auRID[5];	RID
uchar	auSN[3];	Serial number of certification

## EMV\_ticCLog

Structure Name	IC Card Transaction log	
Type	Parameter Name	Parameter Specification
uchar	ucAmtFlg;	If Amount Authorised existed? 0-No 1-Yes
uchar	auAmount[6];	Amount, Authorised, BCD encoded.
uchar	ucAmtOthFlg;	If Amount other existed? 0-No 1-Yes
uchar	auAmountOth[6];	Amount Other, BCD encoded.
uchar	ucDateFlg;	If Transaction Date existed? 0-No 1-Yes
uchar	auDate[3];	Transaction Date, Format as YYMMDD, BCD encoded.
uchar	ucTimeFlg;	If Transaction Time existed? 0-No 1-Yes
uchar	auTime[3];	Transaction Time, Format as HHMMSS, BCD encoded.
uchar	ucCntCFlg;	If Cuntry Code existed? 0-No 1-Yes
uchar	auCntCode[2];	Cuntry Code(9F1A)
uchar	ucCurCFlg;	If Currency Code existed? 0-No 1-Yes
uchar	auCurCode[2];	Currency Code(5F2A)
uchar	ucATCFlg;	If ATC existed? 0-No 1-Yes
uchar	auATC[2];	ATC (9F36)
uchar	ucSevFlg;	If Service Type existed? 0-No 1-Yes
uchar	ucServeType;	Service Type(9C)
uchar	ucMchFlg;	If Merchant Name existed? 0-No 1-Yes
char	szMchName[30];	Merchant Name(9F4E)
uchar	ucTLVLen;	TLVlist Length
uchar	auTLV[256];	TLVlist, stored other TLV elements which without listed above.

## EMV\_tECCLog

Structure Name	IC Card load Transaction log	
Type	Parameter Name	Parameter Specification
uchar	<b>auECTag[2];</b>	Tag name of IC Card offline Balance.
uchar	<b>auPreValue[6];</b>	Balance value before modify
uchar	<b>auAftValue[6];</b>	Balance value after modify
uchar	<b>ucDateFlg;</b>	If Transaction Date existed? 0-No 1-Yes
uchar	<b>auDate[3];</b>	Transaction Date, Format as YYMMDD, BCD encoded.
uchar	<b>ucTimeFlg;</b>	If Transaction Time existed? 0-No 1-Yes
uchar	<b>auTime[3];</b>	Transaction Time, Format as HHMMSS, BCD encoded.
uchar	<b>ucCntCFlg;</b>	If Country Code existed? 0-No 1-Yes
uchar	<b>auCntCode[2];</b>	Country Code(9F1A)
uchar	<b>ucATCFlg;</b>	If ATC existed? 0-No 1-Yes
uchar	<b>auATC[2];</b>	ATC (9F36)
uchar	<b>ucMchFlg;</b>	If Merchant Name existed? 0-No 1-Yes
char	<b>szMchName[30];</b>	Merchant Name(9F4E)
uchar	<b>ucTLVLen;</b>	TLVlist Length
uchar	<b>auTLV[50];</b>	TLVlist, store other TLV elements not listed above.



## EMV\_tCandAIDInfo

Structure Name	Candidate AID information.	
Type	Parameter Name	Parameter Specification
uchar	ucAIDLen;	Length of AID
uchar	auAID[EMV_LEN_MAX_AID];	AID
uchar	ucLabelLen;	Length of Application Label
uchar	auAppLabel[16];	Application Label
uchar	ucAPNLen;	Length of Application Preferred Name
uchar	auAPN[16];	Application Preferred Name
uchar	ucAPIIDFlag;	If Application priority existed? 0-No 1-Yes
uchar	ucAPIID;	Application priority
uchar	ucLangPrefLen;	Length of Prefer language list
uchar	auLangPref[8];	Prefer language list
uchar	ucIssCTIndexFlag;	If Issuer Code Table Index existed? 0-No 1-Yes
uchar	ucIssCTIndex;	Issuer Code Table Index
uchar	ucKernelIDLen;	Kernel ID length
uchar	auKernelID[8];	kernel ID(9F2A)

## EMV\_tAIDCandList

Structure Name	Candidate AID information list	
Type	Parameter Name	Parameter Specification
uchar	<b>ucReSelectFlag;</b>	If Re do Application selection flag? 0-No 1-Yes
uchar	<b>ucCandAIDNum;</b>	The Number of candidate AID
EMV_tCandAIDInfo	<b>patCandList</b>	Candidate AID information list.

## EMV\_tFinalData

Structure Name	Final selection AID information	
Type	Parameter Name	Parameter Specification
uchar	ucAIDLen;	Final selected AID length
uchar	auAID[EMV_LEN_MAX_AID];	Final selected AID Data
uchar	ucKernelID;	Macro define Kernel ID, Refer to: <a href="#">Macro Definition</a>
uchar	ucPIDLen;	Program ID length
uchar	auPID[16];	Program ID(9F5A), only for VISA

## EMV\_tRecordData

Structure Name	Read ICC Record Return Data	
Type	Parameter Name	Parameter Specification
uchar	ucAIDLen;	AID length
uchar	auAID[EMV_LEN_MAX_AID];	AID
uchar	ucPanLen;	PAN length
uchar	auPan[10];	PAN, BCD encoded.
uchar	ucPanSNFlag;	If PANSequenceNumber existed? 0- No 1-Yes
uchar	ucPanSN;	PANSequenceNumber
uchar	auExpiry[4];	Application Expiration Date, Format as YYYYMMDD BCD encoded.
uchar	ucAlgorithmID;	ODA Algorithm ID: 00-RSA 01-SM
uchar	ucPubKIndex;	CA Public key index.
uchar	ucFlowType;	Flow type of transaction. Refer to: <a href="#">Macro Definition</a>
uchar	auECIAC[6];	ECash transaction issuer authorized code.
uchar	ucSFI11;	SFI existed flag, 0-No 1-Yes Only for flow of VISA Paywave2
uchar	RFULen;	Length of RFU
uchar	RFU[256];	RFU

## EMV\_tCVM

Structure Name	Cardholder Verification Method	
Type	Parameter Name	Parameter Specification
uchar	<b>ucCVM;</b>	CVM Flag, Refer to: <a href="#">Macro Definition</a>
uchar	<b>ucPINTimes;</b>	For offline PIN CVM: PIN enter remaining times. If 0 means ICC didn't return.
uchar	<b>ucCertType;</b>	For Certificate CVM: Certificate type. (Certificate Type: 00-Identity Card 01-certificate of officer 02-passport 03-entry permit 04-temporary Identity card 05- others)
uchar	<b>ucCertLen;</b>	For Certificate CVM: length of certificate number
uchar	<b>auCert[40];</b>	For Certificate CVM: Certificate number

## EMV\_tDisplayMsg

Structure Name	Display Message Information	
Type	Parameter Name	Parameter Specification
uchar	ucMsgID;	Message ID, Refer to: <a href="#">Macro Definition</a>
uchar	ucCurrency ;	Currency flag: 0-RMB 1-Dollar
uchar	ucDataLen;	Message length
uchar	auData[30];	Message content

## EMV\_tErrorID

Structure Name	EMV Error Location	
Type	Parameter Name	Parameter Specification
uchar	ucL1;	Level 1 error location , Refer to: <a href="#">Macro Definition</a>
uchar	ucL2;	Level2 error location,Refer to: <a href="#">Macro Definition</a>
uchar	ucL3;	Level3 error location, Refer to: <a href="#">Macro Definition</a>
uchar	ucMsgID;	Message ID, Refer to: <a href="#">Macro Definition</a>
ushort	usSW12;	APDUStatus Code.

## EMV\_tTransData

Structure Name	Transaciton Result Data	
Type	Parameter Name	Parameter Specification
uchar	<b>ucACType;</b>	Application Cryptograph: 0- Declined 1- Approved 2- Online Requested
uchar	<b>ucCVM;</b>	CVM Flag, Refer to: <a href="#">Macro Definition</a>
uchar	<b>ucPanLen;</b>	Length of PAN
uchar	<b>auPan[10];</b>	PAN
uchar	<b>ucPanSnFlag;</b>	If PanSn existed? 0-No 1-Yes
uchar	<b>ucPanSn;</b>	PAN Sequence Number
uchar	<b>auExpiry[3];</b>	Application Expiry
uchar	<b>ucFlowType;</b>	Transaction Flow, Refer to: <a href="#">Macro Definition</a>
uchar	<b>ucECIACFlag;</b>	If ECIAC existed? 0-No 1-Yes
uchar	<b>auECIAC[6];</b>	ECash transaction issuer authorized code.
uchar	<b>ucBalanceFlag;</b>	If balance existed? 0-No 1-Yes
uchar	<b>auBalance[6];</b>	ICC Offline Balance, BCD encoded.
uchar	<b>ucCID;</b>	Cryptograph Information Data
uchar	<b>ucSRLength;</b>	Length of Stript Result, 0-Did't executed issuer script
uchar	<b>auScriptResult[100];</b>	Issuer Script Result.
uchar	<b>ucMSDT1Len;</b>	Track 1 of MSDT1Data.
uchar	<b>auMSDT1Data[200];</b>	Track 1 of Magnetic Stripe Data
uchar	<b>ucMSDT2Len;</b>	Length of MSDT2Data
uchar	<b>auMSDT2Data[100];</b>	Track 2 of Magnetic Stripe Data
uint	<b>uiTLVLen;</b>	Length of TLV list.
uchar	<b>auTLVData[300];</b>	TLV list.



## EMV\_EXPAND\_BASEFUN

Structure Name	Callback Function List from local driver API	
Type	Parameter Name	Parameter Specification
Callback	EX_API_ucICCIO	Reader executive exchange APDU data with card
Callback	EX_API_ucListener	Kernel executive to listen data
Callback	EX_API_ucGetRandomData	Kernel get a random data
Callback	EX_API_ucRSACal	Executive RSA calculation
Callback	EX_API_ucCalculateHash	Executive Hash/SHA1 calculation
Callback	EX_API_ucCalculateHash_SM	Executive Hash_SM calculation
Callback	EX_API_ucVerifySign_SM	Executive Verify Signature_SM
Callback	EX_API_ucVerifyOfflinePIN	Executive Verify OfflinePIN

### Callback Functions Prototype:

```
uchar (*EX_API_ucICCIO)(EMV_tICCDev tICCDevice, uint uiSendLen, const void* pvDataIn, uint *puiRecLen, void* pvDataOut);

uchar (*EX_API_ucListener)(uchar ucFlag, EMVHandle hMagDevice); //Flag:1-Magstripe 2-Contact Chip

uchar (*EX_API_ucGetRandomData)(uint uiLen, uchar *pauData);

uchar (*EX_API_ucRSACal)(uint uiLen, const uchar *pauInData, const EMV_tPKFILESTRU *ptPubKey, uint *puiOutLen, uchar *pauOutData);

uchar (*EX_API_ucCalculateHash)(uchar ucHashFlag, uint uiLen, const uchar *pauInData, uchar *pauHash);

uchar (*EX_API_ucCalculateHash_SM)(uint uiLen, const uchar *pauInData, const EMV_tPKFILESTRU *ptPubKey, uchar *pauHash);

uchar (*EX_API_ucVerifySign_SM)(const uchar *pauHash, uint uiLen, const uchar *pauInData, const EMV_tPKFILESTRU *ptPubKey);

uchar (*EX_API_ucVerifyOfflinePIN)(uchar ucFlag, const uchar *pauRandom, const EMV_tPKFILESTRU *ptPubKey, ushort *pauSW12); //Flag:0-Plaintext Offline PIN 1-Enciphered offline PIN
```

Refer to: [EMV Configuration](#).

## EMV\_EXPAND\_INTERFACE

Structure Name	Callback Function List	
Type	Parameter Name	Parameter Specification
Callback	EXEP_ucWaitCard	Requesting cardholder to show contactless card.
Callback	EXEP_ucAppSelection	Requesting cardholder to select IC card applicaton.
Callback	EXEP_ucFinalSlt	Supply a time slot for application to adjust parameter according to finial selected AID.
Callback	EXEP_ucReadRecord	Supply a time slot for application to deal with messages form card records.
Callback	EXEP_ucCardHolderVerify	Requesting cardholder to make verification according to CVM which is indicated by EMV kernel.
Callback	EXEP_ucOnlineProcess	Requesting online authorization.
Callback	EXEP_vEndProcess	Inform transaction is finished.
Callback	EXEP_vSendOut	Send out some messages classify by kernel instruction (refer to: <a href="#">Macro Definition</a> ) from EMV kernel. (Response not needed)
Callback	EXEP_vObtain	Obtain some messages from application by EMV kernel. (Need response by application)

### Callback Functions Prototype:

```
//Requesting cardholder to show contactless card.
uchar (*EXEP_ucWaitCard)(uchar ucFlag);

//Requesting cardholder to select IC card applicaton.
uchar (*EXEP_ucAppSelection)(const EMV_tAIDCandList *ptDCandList,EMV_tSelectAID *ptSelectedAID4);

//Supply a time slot for application to adjust parameter according to finial selected AID.
uchar (*EXEP_ucFinalSlt)(const EMV_tFinalData *ptFinalData,EMV_tGPOParam *ptDataBack5);

//Supply a time slot for application to deal with messages form card records.
uchar (*EXEP_ucReadRecord)(const EMV_tRecordData *ptRecordData,EMV_tTRMManage *ptTRMManage6);

//Requesting cardholder to make verification according to CVM which is indicated by EMV kernel.
uchar (*EXEP_ucCardHolderVerify)(const EMV_tCVM *ptCVM,EMV_tCHVData *ptCHVData7);

//Requesting online authorization.
uchar (*EXEP_ucOnlineProcess)(const EMV_tTransData *ptTransData,EMV_tHostData *ptHostData8);

//Inform transaction is finished.
void (*EXEP_vEndProcess)(uint uiResult, const EMV_tTransData *ptTransData);

//Send out some messagesclassifiy by kernel instruction from EMV kernel. (Response not needed)
void (*EXEP_vSendOut)(uchar ucINS, uint uiDataLen, const uchar *pauData);

//Obtain some messages from application by EMV kernel. (Need response by application)
void (*EXEP_vObtain)(uchar ucINS, uint uiDataLen, const uchar *pauData);
```

### Specification:

<sup>4</sup>Multiple Application Object have deleted it.

<sup>5</sup>Multiple Application Object have deleted it.

<sup>6</sup>Multiple Application Object have deleted it.

<sup>7</sup>Multiple Application Object have deleted it.

<sup>8</sup>Multiple Application Object have deleted it.

In the callback function list above, when implementing these callback functions, the parameters rendered **Green** are input parameters for the callback functions. Application should store these parameters and return success without any redundant operation. Refer to: [EMV Configuration](#).

~~The parameters rendered **Yellow** are only used for reference when application developer programmes.<sup>9</sup>~~

#### **Program Guide:**

EMV\_EXPAND\_INTERFACE is mandatory for starting a transaction. Each element of this structure is a pointer of call back function. Application will implement these call back functions without any delay and must be non-block.

For example, when EMV Kernel generates an event marking with R1, the Implement of callback function should be simple and follow these steps:

- 1.Store the parameters which return with event R1.
- 2.Set the event flag with R1.
- 3.Return success.

#### **Program example:**

```
uchar EG_ucSignal;           //event flag
uchar EG_auSignalData[1000]; //event parameter
uint EG_uiSignalDataLen;     //length of event parameter
```

//EXEP\_ucFinalISItcallback implementation as follow:

```
uchar EXEP_ucReadRecord(const EMV_tRecordData *ptRecordData, EMV_tTRManage *ptTRManage
{
    EG_uiSignalDataLen=sizeof(EMV_tRecordData);
    memcpy(EG_auSignalData, ptRecordData, EG_uiSignalDataLen);
    EG_ucSignal=SIGNAL_READRECORD;
    return0;
}
```

---

<sup>9</sup>Multiple Application Object have deleted it.

## 8. Obtain Kernel Debug log

During the development or maintenance, if there is any problem caused by EMV kernel, application developer should attempt to obtain debug log from EMV kernel, and then send the email attached description of issues and the kernel debug log to EMV expert, which are great helpful for expert to analyze the reason of issue. The following indicate the steps of how to obtain debug log from EMV kernel:

1. Application should call the API of **EMV\_Kern\_vSwitchDebug** to active the function of debug log writing before starting a transaction. There are three kinds of debug mode. Refer to [API specification](#). Choose 1 or 2 will active the function of debug log.
2. If debug mode is 1, obtaining debug log after transaction finished, EMV kernel will send out the debug log after a transaction finished or terminated by callback function of EXEP\_vSendOut with INS marking EMV\_INS\_DBLOG. All of messages of debug log encoded with ASCII. Application should store the messages through writing file or sending to computer and receiving with tools on computer.
3. If debug mode is 2, obtaining debug log during transaction processing, EMV kernel will send out the debug log during the transaction processing by callback function of EXEP\_vSendOut with INS marking EMV\_INS\_DBLOG. All of messages of debug log encoded with ASCII. Application should store the messages through writing file or sending to computer and receiving with tools on computer.

## Annex A TAG Dictionary

### A.1 TAG of EMV\_KERNELID\_EMV

Kernel ID	EMV_KERNELID_PBOC		
Tag Name	Definition	Description	
EMV_TAG_TM_TERMTYPE	<b>Tag: 9F35</b> <b>Length: 1</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Terminal Type Indicates the environment of the terminal, itscommunications capability, and its operationalcontrol.	
		Environment	Operator
			Financial Institution
		<b>Attendant:</b> Online only Online and Offline Offline only	Mechant 21 22 23
			Cardholder
			- - -
EMV_TAG_TM_CAP	<b>Tag: 9F33</b> <b>Length: 3</b> <b>Format:b</b> <b>Kernel: EMV</b>	<b>Self-help:</b> Online only Online and Offline Offline only	
		14	24
		15	25
EMV_TAG_TM_CAP_AD	<b>Tag: 9F40</b> <b>Length: 5</b> <b>Format:b</b> <b>Kernel: EMV</b>	Terminal Capabilities Indicates the card data input, CVM, andsecurity capabilities of the terminal	
		Additional Terminal Capabilities Indicates the data input and outputcapabilities of the terminal	
EMV_TAG_TM_CNTRYCODE	<b>Tag: 9F1A</b> <b>Length: 2</b> <b>Format:b</b> <b>Kernel: EMV</b>	Terminal Country Code Indicates the country of the terminal,represented according to ISO 3166	
EMV_TAG_TM_CURCODE	<b>Tag: 5F2A</b> <b>Length: 2</b> <b>Format:b</b> <b>Kernel: EMV</b>	TransactionCurrency Code Indicates the currency code of the transactionaccording to ISO 4217	
EMV_TAG_TM_FLOORLMT	<b>Tag: 9F1B</b> <b>Length: 4</b> <b>Format:b</b> <b>Kernel: EMV</b>	Terminal Floor Limit Indicates the floor limit in the terminal inconjunction with the AID	
EMV_TAG_TM_AID	<b>Tag: 9F06</b> <b>Length: 5-16</b> <b>Format:b</b>	ApplicationIdentifier (AID) –terminal Identifies the application as described inISO/IEC 7816-5	

	<b>Kernel: EMV</b>	
<b>EMV_TAG_TM_AUTHAMNTN</b>	<b>Tag: 9F02</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Amount, Authorised (Numeric) Authorised amount of the transaction(excluding adjustments)
<b>EMV_TAG_TM_OTHERAMNTN</b>	<b>Tag: 9F03</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Amount, Other (Numeric) Secondary amount associated with the transaction representing a cashback amount
<b>EMV_TAG_TM_TRANSDATE</b>	<b>Tag: 9A</b> <b>Length: 3</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Transaction Date Local date that the transaction was authorized, format as YYMMDD
<b>EMV_TAG_TM_TRANSTIME</b>	<b>Tag: 9F21</b> <b>Length: 3</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Transaction Time Local time that the transaction was authorized, format as HHMMSS
<b>EMV_TAG_TM_TRSEQCNTR</b>	<b>Tag: 9F41</b> <b>Length:2- 4</b> <b>Format:BCD</b> <b>Kernel: EMV</b>	Transaction Sequence Counter Counter maintained by the terminal that is incremented by one for each transaction
<b>EMV_TAG_TM_ARC</b>	<b>Tag: 8A</b> <b>Length: 2</b> <b>Format:an</b> <b>Kernel: EMV</b>	Authorisation Response Code Code that defines the disposition of a message
<b>EMV_TAG_TM_AUTHCODE</b>	<b>Tag: 89</b> <b>Length:6</b> <b>Format:b</b> <b>Kernel: EMV</b>	Authorisation Code Value generated by the authorisation authority for an approved transaction
<b>EMV_TAG_TM_APPVERNO</b>	<b>Tag: 9F09</b> <b>Length:2</b> <b>Format:b</b> <b>Kernel: EMV</b>	Version number assigned by the Issuer for the application. For AMEX3.1 this specification the Application Version Number must always be '0001'.
<b>EMV_TAG_TM_TRANSTYPE</b>	<b>Tag: 9C</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: EMV</b>	Transaction type: 0x00:Goods/Service 0x09:CashBack 0x01:Cash 0x20:Refund
<b>EMV_TAG_TM_ACQID</b>	<b>Tag: 9F01</b> <b>Length:6</b> <b>Format:n6-11</b> <b>Kernel: EMV</b>	Uniquely identifies the acquirer within each payment system.

<b>EMV_TAG_TM_MCHNAMELOC</b>	<b>Tag: 9F4E</b> <b>Length:Var.</b> <b>Format:ans</b> <b>Kernel: EMV</b>	Indicates the name and location of the merchant
<b>EMV_TAG_TM_MCHCATCODE</b>	<b>Tag: 9F15</b> <b>Length:2</b> <b>Format:n4</b> <b>Kernel: EMV</b>	Classifies the type of business being done by the merchant, represented according to ISO 8583:1993 for Card Acceptor Business Code

## A.2 TAG of EMV\_KERNELID\_PBOC

Kernel ID	EMV_KERNELID_PBOC																																						
Tag Name	Definition	Description																																					
C_TAG_TM_9F7A	<b>Tag:</b> 9F7A <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> PBOC	Indicate supporting Ecash or not. 0 - No 1 - Yes																																					
C_TAG_TM_DF69	<b>Tag:</b> DF69 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> PBOC	Indicate supporting PBOC SM algorithm or not. 0 - No 1 - Yes																																					
C_TAG_TM_9F66	<b>Tag:</b> 9F66 <b>Length:</b> 4 <b>Format:</b> b <b>Kernel:</b> PBOC	TerminalTransaction Qualifiers Indicate terminal capabilities, requirements, and preferences to the card. <table border="1"> <thead> <tr> <th>字节</th><th>位</th><th>定义</th></tr> </thead> <tbody> <tr> <td rowspan="8">1</td><td>8</td><td>预留</td></tr> <tr> <td>7</td><td>1 – 支持非接触式借记/贷记应用 0 – 不支持非接触式借记/贷记应用</td></tr> <tr> <td>6</td><td>1 – 支持 qPBOC 0 – 不支持 qPBOC</td></tr> <tr> <td>5</td><td>1 – 支持接触式借记/贷记应用 0 – 不支持接触式借记/贷记应用</td></tr> <tr> <td>4</td><td>1 – 终端仅支持脱机 0 – 终端具有联机能力</td></tr> <tr> <td>3</td><td>1 – 支持联机 PIN 0 – 不支持联机 PIN</td></tr> <tr> <td>2</td><td>1 – 支持签名 0 – 不支持签名</td></tr> <tr> <td>1</td><td>预留</td></tr> <tr> <td rowspan="3">2</td><td>8</td><td>1 – 要求联机密文 0 – 不要求联机密文</td></tr> <tr> <td>7</td><td>1 – 要求 CVM 0 – 不要求 CVM</td></tr> <tr> <td>6-1</td><td>预留</td></tr> <tr> <td>3</td><td>8-1</td><td>预留</td></tr> <tr> <td rowspan="2">4</td><td rowspan="2">8</td><td>1 – 终端支持“01”版本的 iDDA</td></tr> <tr> <td>0 – 终端仅支持“00”版本的 iDDA</td></tr> <tr> <td></td><td>7-1</td><td>预留</td></tr> </tbody> </table>	字节	位	定义	1	8	预留	7	1 – 支持非接触式借记/贷记应用 0 – 不支持非接触式借记/贷记应用	6	1 – 支持 qPBOC 0 – 不支持 qPBOC	5	1 – 支持接触式借记/贷记应用 0 – 不支持接触式借记/贷记应用	4	1 – 终端仅支持脱机 0 – 终端具有联机能力	3	1 – 支持联机 PIN 0 – 不支持联机 PIN	2	1 – 支持签名 0 – 不支持签名	1	预留	2	8	1 – 要求联机密文 0 – 不要求联机密文	7	1 – 要求 CVM 0 – 不要求 CVM	6-1	预留	3	8-1	预留	4	8	1 – 终端支持“01”版本的 iDDA	0 – 终端仅支持“00”版本的 iDDA		7-1	预留
字节	位	定义																																					
1	8	预留																																					
	7	1 – 支持非接触式借记/贷记应用 0 – 不支持非接触式借记/贷记应用																																					
	6	1 – 支持 qPBOC 0 – 不支持 qPBOC																																					
	5	1 – 支持接触式借记/贷记应用 0 – 不支持接触式借记/贷记应用																																					
	4	1 – 终端仅支持脱机 0 – 终端具有联机能力																																					
	3	1 – 支持联机 PIN 0 – 不支持联机 PIN																																					
	2	1 – 支持签名 0 – 不支持签名																																					
	1	预留																																					
2	8	1 – 要求联机密文 0 – 不要求联机密文																																					
	7	1 – 要求 CVM 0 – 不要求 CVM																																					
	6-1	预留																																					
3	8-1	预留																																					
4	8	1 – 终端支持“01”版本的 iDDA																																					
		0 – 终端仅支持“00”版本的 iDDA																																					
	7-1	预留																																					
C_TAG_TM_9F7B	<b>Tag:</b> 9F7B <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> PBOC	Ecash Floor limit If authorized amount higher than Ecash floor limit, transaction will request online authorization.																																					
C_TAG_TM_TRANS_LIMIT	<b>Tag:</b> DF8124 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> PBOC	Contactless Transaction Limit If authorized amount higher than Transaction limit, transaction will be terminated.																																					
C_TAG_TM_CVM_LIMIT	<b>Tag:</b> DF8126 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> PBOC	Contactless CVM Required Limit If authorized amount higher than CVM limit, transaction will be requested CVM.																																					
C_TAG_TM_FLOOR_LIMIT	<b>Tag:</b> DF8123 <b>Length:</b> 6	Contactless Floor Limit If authorized amount higher than floor limit, transaction																																					



	<b>Format:</b> BCD <b>Kernel:</b> PBOC	will be requested online authorizaton.
<b>C_TAG_TM_RD_RCP</b>	<b>Tag:</b> DF06 <b>Length:</b> 2 <b>Format:</b> b <b>Kernel:</b> PBOC	Reader configuration parameters B1b8: Status Check enabled/disabled (1b= enabled and 0b= disabled) B1b7: Amount, Authorized of Zero Check enabled/disabled (1b= enabled and 0b= disabled) B1b6: Amount, Authorized of Zero Option (1b=Option 1 and 0b=Option 2, this bit is only applicable when the reader is online-capable) B1b5: Reader Contactless Transaction Limit Check enabled/disabled (1b= enabled and 0b= disabled) B1b4: Reader CVM Required Limit Check enabled/disabled (1b= enabled and 0b= disabled) B1b3: Reader Contactless Floor Limit Check enabled/disabled (1b= enabled and 0b= disabled) B1b2: Exception file enabled/disabled (1b= enabled and 0b= disabled) All other bits are RFU"

### A.3 TAG of EMV\_KERNELID\_VISA

Kernel ID	EMV_KERNELID_VISA	
Tag Name	Definition	Description
V_TAG_RD_RCP	<b>Tag:</b> DF06 <b>Length:</b> 2 <b>Format:</b> b <b>Kernel:</b> VISA	<p>Reader configuration parameters</p> <p>B1b8: Status Check enabled/disabled (1b= enabled and 0b= disabled)            B1b7: Amount, Authorized of Zero Check enabled/disabled (1b= enabled and 0b= disabled)            B1b6: Amount, Authorized of Zero Option (1b=Option 1 and 0b=Option 2, this bit is only applicable when the reader is online-capable)            B1b5: Reader Contactless Transaction Limit Check enabled/disabled (1b= enabled and 0b= disabled)            B1b4: Reader CVM Required Limit Check enabled/disabled (1b= enabled and 0b= disabled)            B1b3: Reader Contactless Floor Limit Check enabled/disabled (1b= enabled and 0b= disabled)            B1b2: Exception file enabled/disabled (1b= enabled and 0b= disabled)            All other bits are RFU"</p> <hr/> <p>B1b1:RFU(VISA-Certification Revocation List)</p>
V_TAG_TM_9F66	<b>Tag:</b> 9F66 <b>Length:</b> 4 <b>Format:</b> b <b>Kernel:</b> VISA	<p>TerminalTransaction Qualifiers</p> <p>Indicate terminal capabilities, requirements, and preferences to the card.</p> <p>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.  <i>Note:</i> Readers compliant to this specification set TTQ byte 1 bit 1 to 0b.</p> <p>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)</p> <p>Byte 3            bit 8: 1 = Issuer Update Processing supported            bit 7: 1 = Mobile functionality supported (Consumer Device CVM)            bits 6-1: RFU (000000)</p> <p>Byte 4            RFU ('00')</p>
V_TAG_TM_TRANS_LIMIT	<b>Tag:</b> DF8124 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> VISA	<p>Contactless Transaction Limit</p> <p>If authorized amount is <b>greater than or equal to</b> the Transaction limit, transaction will be terminated.</p> <p>It is used in conjunction with V_TAG_RD_RCP(DF06).</p>
V_TAG_TM_CVM_LIMIT	<b>Tag:</b> DF8126 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> VISA	<p>Contactless CVM Required Limit</p> <p>If authorized amount is <b>greater than or equal to</b> the CVM limit, transaction will be requested CVM.</p> <p>It is used in conjunction with V_TAG_RD_RCP(DF06).</p>
V_TAG_TM_FLOOR_LIMIT	<b>Tag:</b> DF8123	<p>Contactless Floor Limit</p>

	<p><b>Length: 6</b></p> <p><b>Format:BCD</b></p> <p><b>Kernel: VISA</b></p>	<p>If authorized amount is <b>greater than</b> the floor limit, transaction will be requested online authorizaton.</p> <p>It is used in conjunction with <b>V_TAG_RD_RCP</b>(DF06).</p> <p>If the Amount, Authorized is greater than either the Reader Contactless Floor Limit or (if the Reader Contactless Floor Limit is not present) the applicable Terminal Floor Limit (tag '9F1B'), then the reader shall indicate Online Cryptogram Required (set TTQ byte 2 bit 8 to 1b).</p>
--	---	--

#### A.4 TAG of EMV\_KERNELID\_MASTER

Kernel ID	EMV_KERNELID_MASTER	
Tag Name	Definition	Description
M_TAG_TM_TRANS_LIMIT	<b>Tag: DF8124</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: MASTER</b>	Contactless Transaction Limit If authorized amount higher than Transaction limit, transaction will be terminated.
M_TAG_TM_TRANS_LIMIT_CDV	<b>Tag: DF8125</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: MASTER</b>	Contactless Transaction Limit while CDV support Indicates the transaction amount above which the transaction is not allowed, when on-device cardholder verification is supported.
M_TAG_TM_CVM_LIMIT	<b>Tag: DF8126</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: MASTER</b>	Contactless CVM Required Limit If authorized amount higher than CVM limit, transaction will be requested CVM.
M_TAG_TM_FLOOR_LIMIT	<b>Tag: DF8123</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: MASTER</b>	Contactless Floor Limit If authorized amount higher than floor limit, transaction will be requested online authorization.
M_TAG_TM_9F7C	<b>Tag: '9F7C'</b> <b>Length: 20</b> <b>Format: b</b> <b>Kernel: MASTER</b>	Proprietary merchant data that may be requested by the Card
M_TAG_TM_9F53	<b>Tag: '9F53'</b> <b>Length: 1</b> <b>Format: an</b> <b>Kernel: MASTER</b>	This is a data object defined by MasterCard which indicates the type of transaction being performed, and which may be used in card risk management
M_TAG_TM_9F6D	<b>Tag: '9F6D'</b> <b>Length: 2</b> <b>Format: b</b> <b>Kernel: MASTER</b>	Version number assigned by the payment system for the specific mag-stripe mode functionality of the Kernel.

## A.5 TAG of EMV\_KERNELID\_AMEX

Kernel ID	EMV_KERNELID_AMEX																																																																																											
Tag Name	Definition	Description																																																																																										
A_TAG_TM_9F6D	<b>Tag:</b> 9F6D <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> AMEX	Contactless ReaderCapabilities <table><tr><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>Specification 3.1</td></tr><tr><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td></td></tr><tr><td>0</td><td>0</td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>Expresspay 1.0</td></tr><tr><td>0</td><td>0</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td>RFU</td></tr><tr><td>0</td><td>1</td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>Expresspay 2.0 Magstripe Only, or Expresspay ≥3.0 Magstripe-CVM Not Required</td></tr><tr><td>0</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td>Expresspay ≥3.0 Magstripe-CVM Required</td></tr><tr><td>1</td><td>0</td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>Expresspay 2.0 - EMV and Magstripe</td></tr><tr><td>1</td><td>0</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td>RFU</td></tr><tr><td>1</td><td>1</td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>Expresspay ≥3.0 EMV and Magstripe-CVM Not Required</td></tr><tr><td>1</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td>Expresspay ≥3.0 EMV and Magstripe-CVM Required</td></tr></table> ‘00’ = Expresspay 1.0 ‘40’ = Expresspay 2.0Magstripe only ‘48’ = Expresspay 2.0Magstripe – Mobile CVMRequired ‘80’ = Expresspay 2.0EMV and Magstripe ‘C0’ = ExpresspayMobile (XPM) ‘C8’ = ExpresspayMobile (XPM) - MobileCVM Required	b8	b7	b6	b5	b4	b3	b2	b1	Specification 3.1	↓	↓	↓	↓	↓	↓	↓	↓		0	0			0				Expresspay 1.0	0	0			1				RFU	0	1			0				Expresspay 2.0 Magstripe Only, or Expresspay ≥3.0 Magstripe-CVM Not Required	0	1			1				Expresspay ≥3.0 Magstripe-CVM Required	1	0			0				Expresspay 2.0 - EMV and Magstripe	1	0			1				RFU	1	1			0				Expresspay ≥3.0 EMV and Magstripe-CVM Not Required	1	1			1				Expresspay ≥3.0 EMV and Magstripe-CVM Required
b8	b7	b6	b5	b4	b3	b2	b1	Specification 3.1																																																																																				
↓	↓	↓	↓	↓	↓	↓	↓																																																																																					
0	0			0				Expresspay 1.0																																																																																				
0	0			1				RFU																																																																																				
0	1			0				Expresspay 2.0 Magstripe Only, or Expresspay ≥3.0 Magstripe-CVM Not Required																																																																																				
0	1			1				Expresspay ≥3.0 Magstripe-CVM Required																																																																																				
1	0			0				Expresspay 2.0 - EMV and Magstripe																																																																																				
1	0			1				RFU																																																																																				
1	1			0				Expresspay ≥3.0 EMV and Magstripe-CVM Not Required																																																																																				
1	1			1				Expresspay ≥3.0 EMV and Magstripe-CVM Required																																																																																				
A_TAG_TM_9F6E	<b>Tag:</b> 9F6E <b>Length:</b> 4 <b>Format:</b> b <b>Kernel:</b> AMEX	Enhanced ContactlessReader Capabilities Byte1: b8==AEIPS contact mode supported b7==Expresspay Magstripe Mode supported b6==Expresspay EMV full online mode supported b5==Expresspay EMV partial online mode supported b4==Expresspay Mobile Supported b3-b1==RFU Byte2: b8==Mobile CVM supported b7==Online PIN supported b6==Signature b5==Plaintext Offline PIN b4-b1==RFU Byte3: b8==Terminal is offline only b7==CVM Required b6-b1==RFU Byte4: b8-b1==RFU																																																																																										
A_TAG_TM_TRANS_LIMIT	<b>Tag:</b> DF8124 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> AMEX	Terminal Contactless Transaction Limit If the Amount, Authorized is <b>exceeds</b> the Reader Contactless Transaction Limit, the transaction shall be terminate.																																																																																										
A_TAG_TM_FLOOR_LIMIT	<b>Tag:</b> DF8123	Terminal Contactless Floor Limit																																																																																										

	<b>Length:6</b> <b>Format:BCD</b> <b>Kernel: AMEX</b>	If authorized amount is <b>exceeds</b> floor limit, transaction will be requested online authorization.
<b>A_TAG_TM_CVM_LIMIT</b>	<b>Tag: DF8126</b> <b>Length:6</b> <b>Format:BCD</b> <b>Kernel: AMEX</b>	Terminal CVM Required Limit If the Amount, Authorized is <b>equal to or exceeds</b> the Reader CVM Required Limit, then CVM processing is required
<b>A_TAG_PREAGAIN</b>	<b>Tag: DF8130</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: AMEX</b>	Indicate whether a TryAgain needed or not: 0x00- Not Try Again 0x01- Try Again
<b>A_TAG_TM_IN_CARD_BIN_RA NGE</b>	<b>Tag: DF8127</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: AMEX</b>	Indicate if the CardBin in the white CardBin list or not: 0xA0 - Out of CardBin list, Transaction shall be declined 0x00 - In the CardBin list, Transaction continue process

#### A.6 TAG of EMV\_KERNELID\_DISCOVER

Kernel ID	EMV_KERNELID_DISCOVER	
Tag Name	Definition	Description
<b>D_TAG_TM_RD_RCP</b>	<b>Tag: DF06</b> <b>Length:2</b> <b>Format:b</b> <b>Kernel:</b> <b>DISCOVER</b>	Reader Configuration Parameters Byte1: DF Descriptions (1b=Enabled/Present 0b=Disabled/Not Present) b8==Status Check(Value: 1-Support 0-Not Support) b7== 'Zero Amount Allowed' flag is present or not (Flag: 1-Present 0-Not Present), using it conjunction with B1b6 b6==Zero Amount Allowed (Value: 1-Allowed 0-Not Allowed) b5==Reader Contactless Transaction Limit Check (Flag: 1-Present 0-Not Present) b4==Reader CVM Required Limit Check (Flag: 1-Present 0-Not Present) b3==Reader Contactless Floor Limit Check (Flag: 1-Present 0-Not Present) b2==Exception File (Flag: 1-Enabled 0-Disabled) b1==Certification Revocation List (Flag: 1-Enabled 0-Disabled) Byte2: b8== 'Status Check' flag is present or not (Flag: 1-Present 0-Not Present), using it conjunction with B1b8 b7-b1==<RFU
<b>D_TAG_TM_9F66</b>	<b>Tag: 9F66</b> <b>Length:4</b> <b>Format:b</b> <b>Kernel:</b> <b>DISCOVER</b>	TerminalTransaction Qualifiers Indicate terminal capabilities, requirements, and preferences to the card. Byte1: b8==Magnetic stripe mode supported b7==RFU b6==EMV Mode supported b5==EMV contact chip supported b4==Offline-only reader b3==Online PIN supported b2==Signature supported b1==RFU

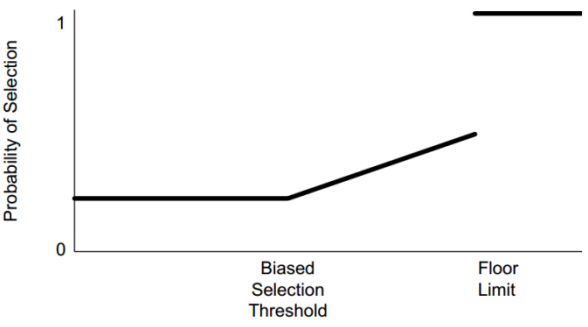
		<p>Byte2:</p> <p>b8==Online Cryptogram required</p> <p>b7==CVM Required</p> <p>b6==(Contact chip) Offline PIN supported</p> <p>b5-b1==RFU</p> <p>Byte3:</p> <p>b8==Issuer Update Processing supported</p> <p>b7==Consumer Device CVM supported</p> <p>b6-b1==RFU</p> <p>Byte4:</p> <p>b8-b1==RFU</p>
<b>D_TAG_TM_TRANS_LIMIT</b>	<p><b>Tag: DF8124</b></p> <p><b>Length:6</b></p> <p><b>Format:BCD</b></p> <p><b>Kernel:</b></p> <p><b>DISCOVER</b></p>	<p>Terminal Contactless Transaction Limit</p> <p>If the Amount, Authorized is <b>equal to or exceeds</b> the Reader Contactless Transaction Limit, the transaction shall be terminate.</p> <p>It is used in conjunction with <b>D_TAG_TM_RD_RCP</b>(DF06).</p>
<b>D_TAG_TM_FLOOR_LIMIT</b>	<p><b>Tag: DF8123</b></p> <p><b>Length:6</b></p> <p><b>Format:BCD</b></p> <p><b>Kernel:</b></p> <p><b>DISCOVER</b></p>	<p>Terminal Contactless Floor Limit</p> <p>If authorized amount is <b>exceeds</b> floor limit, transaction will be requested online authorizaton.</p> <p>It is used in conjunction with <b>D_TAG_TM_RD_RCP</b>(DF06).</p>
<b>D_TAG_TM_CVM_LIMIT</b>	<p><b>Tag: DF8126</b></p> <p><b>Length:6</b></p> <p><b>Format:BCD</b></p> <p><b>Kernel:</b></p> <p><b>DISCOVER</b></p>	<p>Terminal CVM Required Limit</p> <p>If the Amount, Authorized is <b>equal to or exceeds</b> the Reader CVM Required Limit, then CVM processing isrequired.</p> <p>It is used in conjunction with <b>D_TAG_TM_RD_RCP</b>(DF06).</p>

#### A.7 TAG of EMV\_KERNELID\_JCB

Kernel ID	<b>EMV_KERNELID_JCB</b>	
Tag Name	Definition	Description
<b>DEF_TAG_J_COMB_OPTION</b>	<p><b>Tag: DF918404</b></p> <p><b>Length:2</b></p> <p><b>Format:b</b></p> <p><b>Kernel: JCB</b></p>	Reader Configuration Parameters

		<div>Combination Options Byte 1 (Leftmost)</div> <table><tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr><tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>RFU</td></tr><tr><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Status Check supported</td></tr><tr><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td>Offline Data Authentication supported</td></tr><tr><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td>Exception File Check required<sup>11</sup></td></tr><tr><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td>Random Transaction Selection supported</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>0</td><td></td><td></td><td>RFU</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td>EMV Mode Supported<sup>12</sup></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>Legacy Mode Supported<sup>13</sup></td></tr></table> <div>Combination Options Byte 2 (Rightmost)</div> <table><tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr><tr><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>Each bit RFU</td></tr></table>	b8	b7	b6	b5	b4	b3	b2	b1	Meaning	0								RFU		1							Status Check supported			1						Offline Data Authentication supported				1					Exception File Check required <sup>11</sup>					1				Random Transaction Selection supported						0			RFU							1		EMV Mode Supported <sup>12</sup>								1	Legacy Mode Supported <sup>13</sup>	b8	b7	b6	b5	b4	b3	b2	b1	Meaning	0x	0x	0x	0x	0x	0x	0x	0x	Each bit RFU																											
b8	b7	b6	b5	b4	b3	b2	b1	Meaning																																																																																																																								
0								RFU																																																																																																																								
	1							Status Check supported																																																																																																																								
		1						Offline Data Authentication supported																																																																																																																								
			1					Exception File Check required <sup>11</sup>																																																																																																																								
				1				Random Transaction Selection supported																																																																																																																								
					0			RFU																																																																																																																								
						1		EMV Mode Supported <sup>12</sup>																																																																																																																								
							1	Legacy Mode Supported <sup>13</sup>																																																																																																																								
b8	b7	b6	b5	b4	b3	b2	b1	Meaning																																																																																																																								
0x	0x	0x	0x	0x	0x	0x	0x	Each bit RFU																																																																																																																								
DEF_TAG_J_TIP	<div>Tag: DF918408</div> <div>Length:3</div> <div>Format:b</div> <div>Kernel: JCB</div>	<div>Terminal Interchange Profile</div> <div>TIP Byte 1 (Leftmost)</div> <table><tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>CVM required by reader / N/A<sup>14</sup></td></tr><tr><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Signature supported</td></tr><tr><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td>Online PIN supported</td></tr><tr><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td>On-Device CVM supported</td></tr><tr><td></td><td></td><td></td><td></td><td>0</td><td></td><td></td><td></td><td>RFU</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td>Reader is a Transit Reader</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td>EMV contact chip supported</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>(Contact Chip) Offline PIN supported</td></tr></table> <div>TIP Byte 2</div> <table><tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Issuer Update supported<sup>15</sup></td></tr><tr><td></td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>Each bit RFU</td></tr></table> <div>TIP Byte 3 (Rightmost)</div> <table><tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr><tr><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>0x</td><td>Each bit RFU</td></tr></table>	b8	b7	b6	b5	b4	b3	b2	b1	Meaning	1								CVM required by reader / N/A <sup>14</sup>		1							Signature supported			1						Online PIN supported				1					On-Device CVM supported					0				RFU						1			Reader is a Transit Reader							1		EMV contact chip supported								1	(Contact Chip) Offline PIN supported	b8	b7	b6	b5	b4	b3	b2	b1	Meaning	1								Issuer Update supported <sup>15</sup>		0x	0x	0x	0x	0x	0x	0x	Each bit RFU	b8	b7	b6	b5	b4	b3	b2	b1	Meaning	0x	0x	0x	0x	0x	0x	0x	0x	Each bit RFU
b8	b7	b6	b5	b4	b3	b2	b1	Meaning																																																																																																																								
1								CVM required by reader / N/A <sup>14</sup>																																																																																																																								
	1							Signature supported																																																																																																																								
		1						Online PIN supported																																																																																																																								
			1					On-Device CVM supported																																																																																																																								
				0				RFU																																																																																																																								
					1			Reader is a Transit Reader																																																																																																																								
						1		EMV contact chip supported																																																																																																																								
							1	(Contact Chip) Offline PIN supported																																																																																																																								
b8	b7	b6	b5	b4	b3	b2	b1	Meaning																																																																																																																								
1								Issuer Update supported <sup>15</sup>																																																																																																																								
	0x	0x	0x	0x	0x	0x	0x	Each bit RFU																																																																																																																								
b8	b7	b6	b5	b4	b3	b2	b1	Meaning																																																																																																																								
0x	0x	0x	0x	0x	0x	0x	0x	Each bit RFU																																																																																																																								
DEF_TAG_J_TRANS_LIMIT	<div>Tag: DF918402</div> <div>Length:6</div> <div>Format:BCD</div> <div>Kernel: JCB</div>	<div>Terminal Contactless Transaction Limit</div> <div>If the Amount, Authorized is equal to or exceeds the Reader Contactless Transaction Limit, the transaction shall be terminate.</div>																																																																																																																														
DEF_TAG_J_FLOOR_LIMIT	<div>Tag: DF918401</div> <div>Length:6</div> <div>Format:BCD</div> <div>Kernel: JCB</div>	<div>Terminal Contactless Floor Limit</div> <div>If authorized amount greater than floor limit, transaction will be requested online authorizatn.</div>																																																																																																																														
DEF_TAG_J_CVM_LIMIT	<div>Tag: DF918403</div> <div>Length:6</div> <div>Format:BCD</div> <div>Kernel: JCB</div>	<div>Terminal CVM Required Limit</div> <div>If the Amount, Authorized is equal to or exceeds the Reader CVM Required Limit, then CVM processing isrequired.</div>																																																																																																																														
DEF_TAG_J_RS_MAX_PERCENT	<div>Tag: DF918405</div>	<div>Maximum Target Percentage (0-99), larger than Target</div>																																																																																																																														

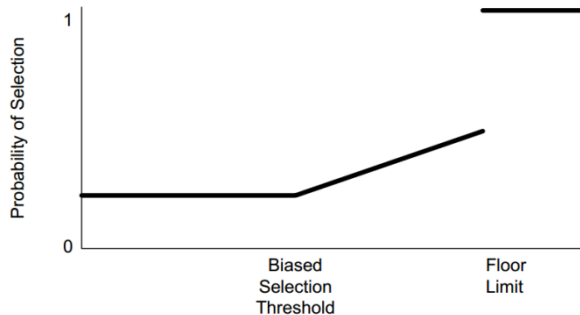


	<b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	Percentages.
<b>DEF_TAG_J_RS_TARGET_PERCENT</b>	<b>Tag: DF918406</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	Target Percentages(0-99)
<b>DEF_TAG_J_RS_THRESH_VALUE</b>	<b>Tag: DF918409</b> <b>Length:4</b> <b>Format:b</b> <b>Kernel: JCB</b>	<p>BiasedSelectionThreshold, if authorized amount higher than BiasedSelectionThreshold, the chance of online authorization will be increase.</p> <p>The relationship between online authorization probability and Floor limit and BiasedSelectionThreshold as follow figure:</p>  <p>Reference: 10.6.2 Random Transaction Selection of EMV specification Book 3.</p>
<b>DEF_TAG_J_ONLINE_TWOPRE</b>	<b>Tag: DF918410</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	<p>Online update script in two ways:</p> <p>present And hold card Or Two present card,Through online transaction data EMV_tTransData in the callback function,andtodetermine whether a label DEF_TAG_J_ONLINE_TWOPRE exist in auTLVData, there are hints to swing card, otherwise their tips to keep spending.</p>

## A.8 TAG of EMV\_KERNELID\_DEFINE

Kernel ID	<b>EMV_KERNELID_DEFINE</b>	
Tag Name	Definition	Description
<b>DEF_TAG_PSE_FLAG</b>	<b>Tag: DF918101</b> <b>Length: 1</b> <b>Format:b</b> <b>Kernel: All</b>	<p>Application selection way:</p> <p>0 - PSE selection first and then AID selection</p> <p>1 - Only PSE selection</p> <p>2 - Only AID selection</p> <p>3 - Only PPSE selection</p> <p>4 - PPSE First, AID selection Second (Discover ZIP Mode)</p>
<b>DEF_TAG_GAC_CONTROL</b>	<b>Tag: DF918102</b>	Generate AC control flag

	<b>Length: 1</b> <b>Format:b</b> <b>Kernel: All</b>	0 - Normal 1 - Force offline 2 - Force online 3 - Force decline
<b>DEF_TAG_QUERY_ICCLOG</b>	<b>Tag: DF918103</b> <b>Length: 1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate if start aICC log query transaction: 0 - No 1 - Yes
<b>DEF_TAG_SERVICE_TYPE</b>	<b>Tag: DF918104</b> <b>Length: 1</b> <b>Format:b</b> <b>Kernel: All</b>	Service Type, Refer to: <a href="#">Macro Definition</a>
<b>DEF_TAG_START_RECOVERY</b>	<b>Tag: DF918105</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: PBOC</b>	Indicate if start a recovery transaction for a torn transaction: 0 –Normal transaction process 1 - Current torn recovery process 2 – All torn recovery process
<b>DEF_TAG_PAN_IN_BLACK</b>	<b>Tag: DF918106</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate if the PAN listed in exception file: 0 - No 1 - Yes
<b>DEF_TAG_ACCUMULATE_AMOUNT</b>	<b>Tag: DF918107</b> <b>Length:6</b> <b>Format:BCD</b> <b>Kernel: All</b>	Serial offline approved amount accumulate for the same PAN.
<b>DEF_TAG_CHV_STATUS</b>	<b>Tag: DF918108</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate the status of operation of cardholder verification: 0- Non execute 1 - Has executed 2 – Executed fail 3 - Exceed pin try times 4 – Bypass pin
<b>DEF_TAG_ONLINE_STATUS</b>	<b>Tag: DF918109</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate the status of online communication: 0 – online success 1 – online failed
<b>DEF_TAG_AUTHORIZE_FLAG</b>	<b>Tag: DF91810A</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate the result online authorization: 0 – Declined 1 – Approved
<b>DEF_TAG_HOST_TLVDATA</b>	<b>Tag: DF91810B</b> <b>Length:Var</b> <b>Format:b</b> <b>Kernel: All</b>	Hostresponse data while online authorization, such as issuer script, which is TLV format.
<b>DEF_TAG_RAND_SLT_THRESHOLD</b>	<b>Tag: DF91810C</b>	BiasedSelectionThreshold, if authorized amount higher

	<b>Length:6</b> <b>Format:BCD</b> <b>Kernel: All</b>	<p>than BiasedSelectionThreshold, the chance of online authorization will be increase.</p> <p>The relationship between online authorization probability and Floor limit and BiasedSelectionThreshold as follow figure:</p>  <p>Reference: 10.6.2 Random Transaction Selection of EMV specification Book 3.</p>
DEF_TAG_RAND_SLT_PER	<b>Tag: DF91810D</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Target Percentages(0-99)
DEF_TAG_RAND_SLT_MAXPER	<b>Tag: DF91810E</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Maximum Target Percentage (0-99), larger than Target Percentages.
DEF_TAG_TAC_DEFAULT	<b>Tag: DF918110</b> <b>Length:5</b> <b>Format:b</b> <b>Kernel: All</b>	Terminal Action Code, For Default
DEF_TAG_TAC_DECLINE	<b>Tag: DF918111</b> <b>Length:5</b> <b>Format:b</b> <b>Kernel: All</b>	Terminal Action Code, For Denial
DEF_TAG_TAC_ONLINE	<b>Tag: DF918112</b> <b>Length:5</b> <b>Format:b</b> <b>Kernel: All</b>	Terminal Action Code, For Online
DEF_TAG_BALANCE_BEF_GAC	<b>Tag: DF918113</b> <b>Length:6</b> <b>Format:BCD</b> <b>Kernel:</b> <b>PBOC\MASTER</b>	ICC balance before Generate AC command.
DEF_TAG_BALANCE_AFT_GAC	<b>Tag: DF918114</b> <b>Length:6</b> <b>Format:BCD</b>	ICC balance After Generate AC command.

	Kernel: PBOC\MASTER															
DEF_TAG_TORN_SUPPORT	Tag: DF918115 Length:1 Format:b Kernel: ALL	Indicate application support torn transaction or not. 0 - No 1 - Yes														
DEF_TAG_M_TRANS_MODE	Tag: DF918201 Length:1 Format:b Kernel: MASTER	Kernel Configuration for Paypass flow 0 - Mag-stripe Flow only 1 - EMV Flow only 2 - Mag-Stripe and EMV Both														
DEF_TAG_M_BALANCE_SUP	Tag: DF918202 Length:1 Format:b Kernel: MASTER	Balance obtain flag, before or after the GENERATE AC: 0 – don't support any one 1 - only support before Generate AC 2 - only support after Generate AC 3 - Support Both														
DEF_TAG_M_TORN_TRANS	Tag: DF918203 Length:1 Format:b Kernel: MASTER	Indicate Kernel supporting tracks torn transactions and recovery or not. 0 – don't support 1-support														
DEF_TAG_M_CDV_SUP	Tag: DF918204 Length:1 Format:b Kernel: MASTER	Card holder device CVM verification for paypass 0-unsupport 1-support														
DEF_TAG_M_REQ_CVM	Tag: DF918205 Length:1 Format:b Kernel: MASTER	CVM Capability – CVM Required Indicates the CVM capability of the Terminal and Reader when the transaction amount is greater than the Reader CVM Required Limit. <table><tr><th colspan="2">CVM Capability – CVM Required</th></tr><tr><td>b8</td><td>Plaintext PIN for ICC verification</td></tr><tr><td>b7</td><td>Enciphered PIN for online verification</td></tr><tr><td>b6</td><td>Signature (paper)</td></tr><tr><td>b5</td><td>Enciphered PIN for offline verification</td></tr><tr><td>b4</td><td>No CVM required</td></tr><tr><td>b3-1</td><td>RFU</td></tr></table>	CVM Capability – CVM Required		b8	Plaintext PIN for ICC verification	b7	Enciphered PIN for online verification	b6	Signature (paper)	b5	Enciphered PIN for offline verification	b4	No CVM required	b3-1	RFU
CVM Capability – CVM Required																
b8	Plaintext PIN for ICC verification															
b7	Enciphered PIN for online verification															
b6	Signature (paper)															
b5	Enciphered PIN for offline verification															
b4	No CVM required															
b3-1	RFU															
DEF_TAG_M_REQ_NOCVM	Tag: DF918206 Length:1 Format:b Kernel: MASTER	CVM Capability – No CVM Required Indicates the CVM capability of the Terminal and Reader when the transaction amount is less than or equal to the Reader CVM Required Limit.														

		<table><tr><th colspan="2">CVM Capability – No CVM Required</th></tr><tr><td>b8</td><td>Plaintext PIN for ICC verification</td></tr><tr><td>b7</td><td>Enciphered PIN for online verification</td></tr><tr><td>b6</td><td>Signature (paper)</td></tr><tr><td>b5</td><td>Enciphered PIN for offline verification</td></tr><tr><td>b4</td><td>No CVM required</td></tr><tr><td>b3-1</td><td>RFU</td></tr></table>	CVM Capability – No CVM Required		b8	Plaintext PIN for ICC verification	b7	Enciphered PIN for online verification	b6	Signature (paper)	b5	Enciphered PIN for offline verification	b4	No CVM required	b3-1	RFU
CVM Capability – No CVM Required																
b8	Plaintext PIN for ICC verification															
b7	Enciphered PIN for online verification															
b6	Signature (paper)															
b5	Enciphered PIN for offline verification															
b4	No CVM required															
b3-1	RFU															
DEF_TAG_M_MAG_REQ_CVM	Tag: DF918207 Length:1 Format:b Kernel: MASTER	<div>Mag-stripe CVM Capability – CVM Required</div> <div>Indicates the CVM capability of the terminal in the case of a mag-stripe mode transaction when the Amount authorized (Numeric) is greater than the Reader CVM Required Limit</div> <table><tr><th colspan="2">Mag-stripe CVM Capability – CVM Required</th></tr><tr><td rowspan="5">b8-5</td><td>CVM</td></tr><tr><td>0000: NO CVM</td></tr><tr><td>0001: OBTAIN SIGNATURE</td></tr><tr><td>0010: ONLINE PIN</td></tr><tr><td>1111: N/A</td></tr><tr><td colspan="2">Other values: RFU</td></tr><tr><td>b4-1</td><td>RFU</td></tr></table>	Mag-stripe CVM Capability – CVM Required		b8-5	CVM	0000: NO CVM	0001: OBTAIN SIGNATURE	0010: ONLINE PIN	1111: N/A	Other values: RFU		b4-1	RFU		
Mag-stripe CVM Capability – CVM Required																
b8-5	CVM															
	0000: NO CVM															
	0001: OBTAIN SIGNATURE															
	0010: ONLINE PIN															
	1111: N/A															
Other values: RFU																
b4-1	RFU															
DEF_TAG_M_MAG_REQ_NOCVM	Tag: DF918208 Length: Format: Kernel: MASTER	<div>Mag-stripe CVM Capability – No CVM Required</div> <div>Indicates the CVM capability of the Terminal in the case of a mag-stripe mode transaction when the Amount authorized (Numeric) is less than or equal to the Reader CVM Required Limit.</div> <table><tr><th colspan="2">Mag-stripe CVM Capability – No CVM Required</th></tr><tr><td rowspan="5">b8-5</td><td>CVM</td></tr><tr><td>0000: NO CVM</td></tr><tr><td>0001: OBTAIN SIGNATURE</td></tr><tr><td>0010: ONLINE PIN</td></tr><tr><td>1111: N/A</td></tr><tr><td colspan="2">Other values: RFU</td></tr><tr><td>b4-1</td><td>RFU</td></tr></table>	Mag-stripe CVM Capability – No CVM Required		b8-5	CVM	0000: NO CVM	0001: OBTAIN SIGNATURE	0010: ONLINE PIN	1111: N/A	Other values: RFU		b4-1	RFU		
Mag-stripe CVM Capability – No CVM Required																
b8-5	CVM															
	0000: NO CVM															
	0001: OBTAIN SIGNATURE															
	0010: ONLINE PIN															
	1111: N/A															
Other values: RFU																
b4-1	RFU															
DEF_TAG_M_MSG_HOLDTIME	Tag: DF918209 Length:3 Format:BCD Kernel: MASTER	<div>Message Hold Time</div> <div>Indicates the default delay for the processing of the next MSGsignal. The Message Hold Time is an integer in units of 100ms.</div>														
DEF_TAG_M_RF_HOLDTIME	Tag: DF91820A Length:1 Format:b Kernel: MASTER	<div>Hold Time Value</div> <div>Indicates the time that the field is to be turned off after the transaction is completed if requested to do so by the cardholder device. The Hold Time Value is in units of 100ms.</div>														

<b>DEF_TAG_D_ISSUERSCRIPT_EXCUTIVE</b>	<b>Tag: DF918215</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate whether a Issuer Script Process needed or not: 0x00- Not Needed(Default) 0x01- Needed It shold be set to 0x01 when Online Process the Issuer return 71 or 72 issuer script.
<b>DEF_TAG_RESELECT_CONDITION</b>	<b>Tag: DF928103</b> <b>Length:5</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate which condition will cause failure on current AID and request application selection on next AID: Byte 1: b8-Final selection response with SW12 unequal 9000 b7-Final selection response with the data didn't include PDOL b6- Final selection response with the data include PDOL but didn't request 9F66. Byte2-Byte5 RFU
<b>DEF_TAG_PPSE_6A82_TURNTO_AIDLIST</b>	<b>Tag: DF918155</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	If PPSE response 6A82, turn to AID list selection. 0x01-indicates turn to AID List Selection when PPSE rutun 6A82 0x00-do Nothing
<b>DEF_TAG_CHECK_CAPK_INDEXLIST</b>	<b>Tag: DF928104</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: PBOC/VISA</b>	When card returned approve, Whether check CAPK index or not before read the last record. 0x00-not check 0x01-check
<b>DEF_TAG_CTL_AS_CB_FLAG</b>	<b>Tag: DF928105</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Whether execute app select callback function or not in contactless transaction 0x00-not execute 0x01- execute
<b>DEF_TAG_OBTAIN_FLAG</b>	<b>Tag: DF928101</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate whether 'OBTAIN' callback function is to be output after ending of a step 0x00-No need 0x01-Executive OBTAIN after end of ODA step
<b>DEF_TAG_OBTAIN_RETURN_DATA</b>	<b>Tag: DF928102</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	The output data of OBTAIN callback function after ending Of ODA step 0x00-ODA success 0x01-ODA failed
<b>DEF_TAG_ONLIE_ODA_FAIL_FLOW_TYPE</b>	<b>Tag: DF918163</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate the transaction is online or declined when online ODA execute fail. 0x00-online 0x01-declined
<b>DEF_TAG_RESULT_CODE</b>	<b>Tag: DF91810F</b> <b>Length:4</b> <b>Format:b</b> <b>Kernel: All</b>	Addition Result Code, refer to Annex C.
<b>DEF_TAG_ALLOW_DUP_ICC_SAMEV</b>	<b>Tag: DF918140</b>	Indicate that if allow ICC element repeated with same

<b>ALUE</b>	<b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	value: 0- Not allow 1- Allow(Except follow tags: 5A, 57,5F24,9F07,9F32,8C, come from PBOC requirements )
<b>DEF_TAG_ERROR_TYPE</b>	<b>Tag: DF91815A</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate which error type when the transaction was terminated. 0x01-Try Another Interface 0x02-Try Another Payment 0x03-Use Another Card

## Annex B Macro Definition

Macro Name	Value	Indication
<b>Signal Type</b>		
EMV_SIGNAL_ACT	0x01	Activate a transaction
EMV_SIGNAL_NEXT	0x02	Inform kernel to continue process
EMV_SIGNAL_STOP	0xF0	Terminate the transaction
EMV_SIGNAL_CLEAN	0xC0	Kernel housekeeping
<b>Kernel ID</b>		
EMV_KERNELID_EMV	0x00	EMV Contact
EMV_KERNELID_EMVCTLess	0x01	EMV Contactless
EMV_KERNELID_MASTER	0x02	Master Card
EMV_KERNELID_VISA	0x03	VISA
EMV_KERNELID_AMEX	0x04	AMEX
EMV_KERNELID_JCB	0x05	JCB
EMV_KERNELID_DISCOVER	0x06	DISCOVER
EMV_KERNELID_PBOC	0x07	PBOC
EMV_KERNELID_NSICC	0xDA	Indonesian
EMV_KERNELID_DEFINE	0xDE	Kernel Defined
<b>Flow Type</b>		
EMV_FLOWTYPE_EMV	0x01	EMV/PBOC Contact Level2
EMV_FLOWTYPE_ECASH	0x03	PBOC ECash
EMV_FLOWTYPE_QPBOC	0x11	qPBOC
EMV_FLOWTYPE_PBOC_CTLESS	0x12	PBOC Contactless Level2
EMV_FLOWTYPE_MSD	0x13	VISA MSD
EMV_FLOWTYPE_MSD_LEGACY	0x14	VISA MSD Legacy
EMV_FLOWTYPE_QVSDC	0x21	VISA qVSDC
EMV_FLOWTYPE_WAVE2	0x22	VISA PayWave2
EMV_FLOWTYPE_M_CHIP	0x31	MASTER Card PayPass-Chip
EMV_FLOWTYPE_M_STRIPE	0x32	MASTER Card PayPass-Stripe
EMV_FLOWTYPE_J_EMV	0x33	JCB EMV Mode
EMV_FLOWTYPE_J_MAG	0x34	JCB Magstripe Mode
EMV_FLOWTYPE_J_LEGACY	0x35	JCB Legacy mode
EMV_FLOWTYPE_A_XP2_MS	0x41	AMEX ExpressPay Card Magstripe Mode
EMV_FLOWTYPE_A_XP2_EMV	0x42	AMEX ExpressPay Card EMV Mode
EMV_FLOWTYPE_A_XPM_MS	0x43	AMEX ExpressPay Mobile Magstripe Mode
EMV_FLOWTYPE_A_XPM_EMV	0x44	AMEX ExpressPay Mobile EMV Mode
EMV_FLOWTYPE_D_DPAS_MS	0x51	Discover D-PAS Magstripe Mode
EMV_FLOWTYPE_D_DPAS_EMV	0x52	Discover D-PAS EMV Mode
EMV_FLOWTYPE_D_ZIP	0x53	Discover ZIP Mode
<b>Service Type</b>		



EMV_SERVETYPE_GOOD	0x00	Goods
EMV_SERVETYPE_SERVICE	0x01	Service
EMV_SERVETYPE_CASH	0x02	Cash
EMV_SERVETYPE_CASHBACK	0x03	Cashback
EMV_SERVETYPE_REFUND	0x04	Refund
<b>CVM Flag</b>		
EMV_CVMFLAG_NOCVM	0x00	No CVM Verification
EMV_CVMFLAG_OFFLINEPIN	0x01	Offline PIN
EMV_CVMFLAG_ONLINEPIN	0x02	Online PIN
EMV_CVMFLAG_SIGNATURE	0x03	Signature
EMV_CVMFLAG_OLPIN_SIGN	0x04	OnlinePIN and Signature
EMV_CVMFLAG_CDV	0x05	Consumer Device Verification(qVSDC/qPBOC)
EMV_CVMFLAG_CCV	0x06	Confirmation Code Verified(PayPass)
EMV_CVMFLAG_CERTIFICATE	0x11	Certificate Verification (Certificate Type: 00-Identity Card 01-certificate of officer 02-passport 03-entry permit 04-temporary Identity card 05- others)
EMV_CVMFLAG_ECASHPIN	0x21	ECash Change PIN
<b>AC Type</b>		
EMV_ACTION_AAC	0x00	Declined
EMV_ACTION_TC	0x01	Approved
EMV_ACTION_ARQC	0x02	Request Online Authorization
<b>Action Flag</b>		
EMV_FLAG_ADD	0x01	Add
EMV_FLAG_DELETE	0x02	Delete
EMV_FLAG_CLEAR	0x03	Clear
<b>Kernel Instruction</b>		
EMV_INS_SET_TORN	0xA1	Send out torn transaction record
EMV_INS_DEL_TORN	0xD1	Inform application to delete torn record expired
EMV_INS_DISPLAY	0xA2	Send out display information, format reference structure: <a href="#">EMV tDisplayMsg</a>
EMV_INS_TLVDATA	0xA3	Send out TLV data
EMV_INS_CLOSERF	0xA4	Inform application to close contactless interface
EMV_INS_DBLOG	0xDB	Send out debug log
EMV_INS_GET_TORN	0xB1	Obtain torn record specified by kernel
EMV_INS_SEND_DISC	0xE1	Send out discretionarydata (PayPass)
EMV_INS_APPSELECT_DATA	0xC1	Send out application selection data
EMV_INS_SET_FAIL_WATER	0xA5	Inform app save fail transaction water
EMV_INS_DEL_FAIL_WATER	0xA6	Inform app delete fail transaction water
<b>Message ID</b>		
EMV_MSGID_CARD_READ_OK	0x17	Read card finished
EMV_MSGID_TRY_AGAIN	0x21	Try again

<b>EMV_MSGID_APPROVED</b>	0x03	Transaction approved
<b>EMV_MSGID_APPROVED_SIGN</b>	0x1A	Transaction approved and requesting signature
<b>EMV_MSGID_DECLINED</b>	0x07	Transaction declined
<b>EMV_MSGID_ERR_OTH_CARD</b>	0x1C	Transaction error, please try other card.
<b>EMV_MSGID_INSERT_CARD</b>	0x1D	Please insert IC card
<b>EMV_MSGID_SEE_PHONE</b>	0x20	Please check cell phone.
<b>EMV_MSGID_AUTH_WAIT</b>	0x1B	Waiting authorization
<b>EMV_MSGID_CLEAR_DISPLAY</b>	0x1E	Clear screen display
<b>EMV_MSGID_ICC_ACCOUNT</b>	0x1F	Icc Account
<b>EMV_MSGID_PCII</b>	0xF1	Display message according PCII
<b>EMV_MSGID_UNMATCH_PAN</b>	0xF2	In current tron recovery process, unmatched pan
<b>EMV_MSGID_READ_CARD_FAIL</b>	0xF3	In All flash tron recovery process, read card fail
<b>EMV_MSGID_ONLINE_ODA_RESULT</b>	0xF4	In OnlineOda process, the result of ODA
<b>Error Location</b>		
<b>EMV_L1_ERR_TIMEOUT</b>	0x01	ICC APDU communicate time out
<b>EMV_L1_ERR_TRANSMISSION</b>	0x02	ICC APDU transmission error
<b>EMV_L1_ERR_PROTOCOL</b>	0x03	ICC transmit protocol error
<b>EMV_L2_ERR_ICC_DATA_MISS</b>	0x01	ICC Data missing
<b>EMV_L2_ERR_CAM_FAILED</b>	0x02	CAM Fail
<b>EMV_L2_ERR_ICC_STATUS</b>	0x03	APDU Status Error
<b>EMV_L2_ERR_PARSING</b>	0x04	ICC data parsing error.
<b>EMV_L2_ERR_MAX_EXCEEDED</b>	0x05	Exceed max limit.
<b>EMV_L2_ERR_ICC_DATA</b>	0x06	ICC data error.
<b>EMV_L2_ERR_MAG_NOT_SUP</b>	0x07	Don't support magnetic
<b>EMV_L2_ERR_NO_PPSE</b>	0x08	Don't support PPSE
<b>EMV_L2_ERR_PPSE_FAULT</b>	0x09	PPSE fault
<b>EMV_L2_ERR_NO_CAND_AID</b>	0x0A	Candidate AID list is empty.
<b>EMV_L2_ERR_TERM_DATA</b>	0x0F	Terminate parameter error.
<b>EMV_L3_ERR_TIMEOUT</b>	0x01	Time out
<b>EMV_L3_ERR_STOP</b>	0x02	Transaction being terminated
<b>EMV_L3_ERR_AMOUNT</b>	0x03	Amount is absent
<b>WaitCard Callback Parameter Flag</b>		
<b>EMV_FLAG_NORMAL</b>	0x00	Normal transaction wait card flag
<b>EMV_FLAG_SHOW_CARD_AGAIN</b>	0x01	Show card again
<b>EMV_FLAG_ISS_SCRIPT_UPDATE</b>	0x02	Show card again, don't show amount
<b>EMV_FLAG_EXECUTE_CDCVM</b>	0x03	CDCVM wasn't execute, prompt related information and show card again



## Annex C Transaction Return Code

The transaction return code (TRC) indicate the current transaction's status and error type, which come out from the first parameter of the transaction finished event callback API "EXEP\_vEndProcess" , as follows:

```
Void (*EXEP_vEndProcess)(uint uiResult, const EMV_tTransData *ptTransData)
```

TRC is divided into three categories:

### 1、 Transaction finished normally.

The only return code that identifies the normal end of the transaction :

**EMV\_RESULT\_NORMAL**

### 2、 Common Error Code

- Code with 2 bytes , format : 0xEX XX
- There are some common features for this kind of error type, which are appearance frequently and independent with transaction flow. These kind of error codes have been defined as macro for being recongnized easily by developer.

### 3、 Other Error Code

- Code with 4 bytes, format : 0xFN XX YY ZZ
- If the fourth byte ZZ ,has a value 0xFX ,then still need a additional result code for help to position an error exactly. This additional result code had been defined as Tag DEF\_TAG\_RESULT\_CODE ( DF91810F ) .

Program example :

```
void EXEP_vEndProcess(uint uiResult, const EMV_tTransData *ptTransData)
```

```

{
    Uint uiLen;
    Uchar auResultCode[10];
    switch(uiResult)
    {
    case EMV_RESULT_NORMAL:
        Log("Transaction finished and the status of transaction refer EMV_tTransData");Break;
    case EMV_RESULT_NOAPP:
        Log("There is no application match both by terminal and card");Break;
    case EMV_RESULT_NOPUBKEY:
        Log("Quick pass transaction, CA public key missing, transaction terminate");Break;
    case EMV_RESULT_EXPIRY:
        Log("Card application had expired, transaction terminate.");Break;
    case EMV_RESULT_STOP:
        Log("Transacton had been stoped manually ");Break;
    case EMV_RESULT_REPOWERICC:
        Log("Please repower the card and try again");Break;
    case EMV_RESULT_REFUSESERVICE:
        Log("IC card refuse the service.");Break;
    case EMV_RESULT_CARDLOCK:
        Log("IC card had been locked(SW=6A81)");Break;
    case EMV_RESULT_APPLOCK:
        Log("Card application had been locked(SW=6283)");Break;
    case EMV_RESULT_EXCEED_CTLMT:
        Log("Amount larger than contactless limit, transaction terminate");Break;
    case EMV_RESULT_APDU_ERROR:
        Log("APDU error, Please try again");Break;
    case EMV_RESULT_APDU_STATUS_ERROR:
        Log("APDU status error , please try again.");Break;
    default:
        uiLen=0;
        memset(auResultCode, 0, sizeof(auResultCode))
        if ((uiResult&0xF0)==0xF0)
            { //need to get additional result codd from DEF_TAG_RESULT_CODE.
                EMV_Kern_uiGetTLV(tEMVObject,DEF_TAG_RESULT_CODE,4,auResultCode, &uiLen);
            }
        if(!uiLen)
        {
            Log("Other error , transaction terminate. Transaction result code:%04X",uiResult); }
        else
        {Log("Other error , transaction terminate. Transaction result code:%04X+%04X",uiResult,
            auResultCode); }
        break;
    }
}

```

Transaction Finished Normal		
<b>EMV_RESULT_NORMAL</b>	0x0000	Transaction completed
Common Error Code		
<b>EMV_RESULT_BUSY</b>	0xEE01	EMV Kernel is busy. Please wait and try again later.
<b>EMV_RESULT_NOAPP</b>	0xEE02	No AID matched between terminal and card.
<b>EMV_RESULT_NOPUBKEY</b>	0xEE03	For qPBOC and qVSDC transaction, Terminal can't supply CA public key Indicated by the card.
<b>EMV_RESULT_EXPIRY</b>	0xEE04	ICC application is expiry.
<b>EMV_RESULT_FLASHCARD</b>	0xEE06	Flash card has occurred.
<b>EMV_RESULT_STOP</b>	0xEE07	EMV kernel is terminated by application
<b>EMV_RESULT_REPOWERICC</b>	0xEE08	Communicate between ICC and Terminal.
<b>EMV_RESULT_REFUSESERVICE</b>	0xEE09	ICC refused to provide the service.
<b>EMV_RESULT_CARDLOCK</b>	0xEE0A	Card locked.(SW=6A81)
<b>EMV_RESULT_APPLOCK</b>	0xEE0B	Application locked.(SW=6283)
<b>EMV_RESULT_EXCEED_CTLMT</b>	0xEE0C	Amount exceed contactless limit
<b>EMV_RESULT_APDU_ERROR</b>	0xEE0D	APDU Exchange error
<b>EMV_RESULT_APDU_STATUS_ERROR</b>	0xEE0E	Card response APDU SW12!=9000
<b>EMV_RESULT_ALL_FLASH_CARD</b>	0xEE0F	Return all flash card to be deal with process
<b>Other undefined codes</b>	XXXXXXX	Other reason cause the transaction terminated.
Other Error Code		
<b>FN XX YY ZZ</b>	If value of ZZ >=0xF0, Need additional result code DF91810F	
<b>DEF_TAG_RESULT_CODE</b>	Additional result code, Hex format, 4Bytes	