

# 联迪 EMV 轻内核应用开发指南

EMV IC 卡支付应用便捷开发

2018/8/1

福建联迪商用设备有限公司

Version: 2.9

# 目录

目录.....	1
1. 概述.....	3
1.1 简介.....	3
1.2 读者.....	3
1.3 编码说明.....	3
1.4 支持的交易类型.....	3
2. 模块层次示意图.....	6
3. 交易时序图.....	7
4. 事件与信号.....	8
5. 交易流程图.....	9
5.1 交易初始化.....	11
5.2 EMV 标准借贷记交易 .....	12
5.3 PBOC 接触式电子现金交易 .....	14
5.4 PBOC 非接标准借贷记交易 .....	16
5.5 PBOC 非接快速支付交易(qPBOC) .....	18
5.6 VISA (PayWave)交易 .....	20
5.7 MASTERCARD(PayPass)交易 .....	23
5.8 AMEX 美国运通交易 .....	26
5.9 DISCOVER 大莱交易 .....	29
5.10 JCB 日财卡交易 .....	32
5.11 RUPAY(qSPARC).....	34
5.12 读卡片日志交易 .....	41
5.13 读卡片余额交易.....	42
5.14 交易结束处理.....	43
6. API 说明 .....	44
6.1 EMV_Kern_uiCreateObject .....	44
6.2EMV_Kern_uiDestroyObject .....	45
6.3EMV_Kern_uiManageAID.....	46
6.4EMV_Kern_uiUpdateCAIndexList .....	47
6.5EMV_Kern_uiSetCAPubKey .....	48
6.6EMV_Kern_uiSetCAPubKey_SM.....	49
6.7EMV_Kern_uiManageRecCert.....	50
6.8EMV_Kern_uiManageDOL.....	51
6.9 EMV_Kern_uiSignalInTLV .....	52
6.10 EMV_Kern_uiSetTLV .....	53
6.11 EMV_Kern_uiSetTLVList .....	54
6.12 EMV_Kern_uiGetTLV .....	57
6.13 EMV_kern_uiGetBalance .....	58
6.14 EMV_kern_uiGetDataAPDU .....	59

6.15 EMV_Kern_uiGetICCLog .....	60
6.16 EMV_Kern_uiGetECCLog .....	61
6.17 EMV_Kern_vSwitchDebug .....	62
6.18 EMV_kern_uiSetHandle .....	63
7. 结构定义 .....	64
EMV_Configuration .....	64
EMV_tPKFILESTRU .....	65
EMV_tPKFILESTRU_SM .....	66
EMV_tSelectAID .....	67
EMV_tRecCert .....	68
EMV_tICCLog .....	69
EMV_tECCLog .....	70
EMV_tCandAIDInfo .....	71
EMV_tAIDCandList .....	72
EMV_tFinalData .....	73
EMV_tRecordData .....	74
EMV_tCVM .....	75
EMV_tDisplayMsg .....	76
EMV_tErrorID .....	77
EMV_tTransData .....	78
EMV_EXPAND_BASEFUN .....	79
EMV_EXPAND_INTERFACE .....	80
8. 获取内核 Debug 日志 .....	82
附录 A 标签定义列表 .....	83
A.1 TAG of EMV_KERNELID_EMV .....	83
A.2 TAG of EMV_KERNELID_PBOC .....	86
A.3 TAG of EMV_KERNELID_VISA .....	88
A.4 TAG of EMV_KERNELID_MASTER .....	90
A.5 TAG of EMV_KERNELID_AMEX .....	91
A.6 TAG of EMV_KERNELID_DISCOVER .....	93
A.7 TAG of EMV_KERNELID_JCB .....	95
A.8 TAG of EMV_KERNELID_DEFINE .....	97
A.9 TAG of EMV_KERNELID_RUPAY .....	103
附录 B 宏定义集合 .....	106
附录 C 交易返回码 .....	109
附录 E 常见问题解答 .....	112
非接交易相关 .....	112
接触交易相关 .....	114

# 1. 概述

## 1.1 简介

轻 EMV L2 内核，以下简称轻内核，区别于传统 EMVL2 内核，通过整体设计，接口优化，将复杂的 EMV 交易流程集成固化在轻内核中，由专业内核开发人员维护。同时轻内核又开放出足够的窗口以实现各类应用开发的个性化和差异化。轻内核是一套更简单，更快捷，更灵活的 EMV 应用开发内核。

## 1.2 读者

本文档是面向 EMV 应用开发人员或者对 EMV 应用开发有兴趣的读者。

## 1.3 编码说明

### BCD 编码

压缩数值型字符编码，

BCD 编码金额：

举例：6 字节交易金额 123.45，BCD 编码形式为：\x00\x00\x00\x01\x23\x45 （金额右对齐）

BCD 编码日期：

举例：2015 年 3 月 10 日，BCD 编码形式为：\x20\x15\x03\x10

BCD 编码时间：

举例：08:10:59，BCD 编码形式为：\x08\x10\x59

### HEX 编码

#### 16 进制数据编码

HEX 编码 AID：

举例：AID: A000000333010101，HEX 编码为：\xA0\x00\x00\x03\x33\x01\x01\x01

## 1.4 支持的交易类型

- EMV Level2 4.3
- VISA PayWave 2.1.3(qVSDC)/2.2
- MasterCard PayPass 3.0
- PBOC3.0 接触式借贷记
- PBOC3.0 非接触式借贷记
- PBOC3.0 电子现金
- PBOC3.0 qPBOC
- American Express ExpressPay 3.1(美国运通 AMEX 3.1)
- Discover® Contactless D-PAS v1.1 & ZIP Payment v3.1.2(大莱 DISCOVER)

- JCB(日財卡)
- RUPAY (qSPARC)

## 1.5 文档导读

阅读本文档能够了解到轻 EMV 内核实现 EMV 交易的基本结构，及其对外所提供的 API 与应用端交互的实现机制，从而让应用开发人员快速入手 EMV 的应用开发。本文档主要包含 8 个部分，以下为 8 个部分的简要描述，开发人员应重点阅读第 4 部分事件与信号和第 5 部分交易流程图。

第 2 部分模块层次示意图，介绍轻内核与上层应用和底层驱动的对关系。

第 3 部分交易时序图，介绍轻内核实现 EMV 交易的流程以及所产生事件，事件处理信号的时序。

第 4 部分事件与信号，详细介绍了轻内核在各类交易中，所有可能产生的事件的名称，对应的回调函数，期待应用反馈的信号和数据。

第 5 部分交易流程图，详细介绍了在轻内核在处理不同的交易流程所产生的不同的事件以及应用端处理完事件后需要反馈给内核的数据。

第 6 部分 API 说明，详细介绍了内核所提供 API 的使用说明。

第 7 部分结构定义，详细介绍了内核所有输出参数结构的每个成员的意义。

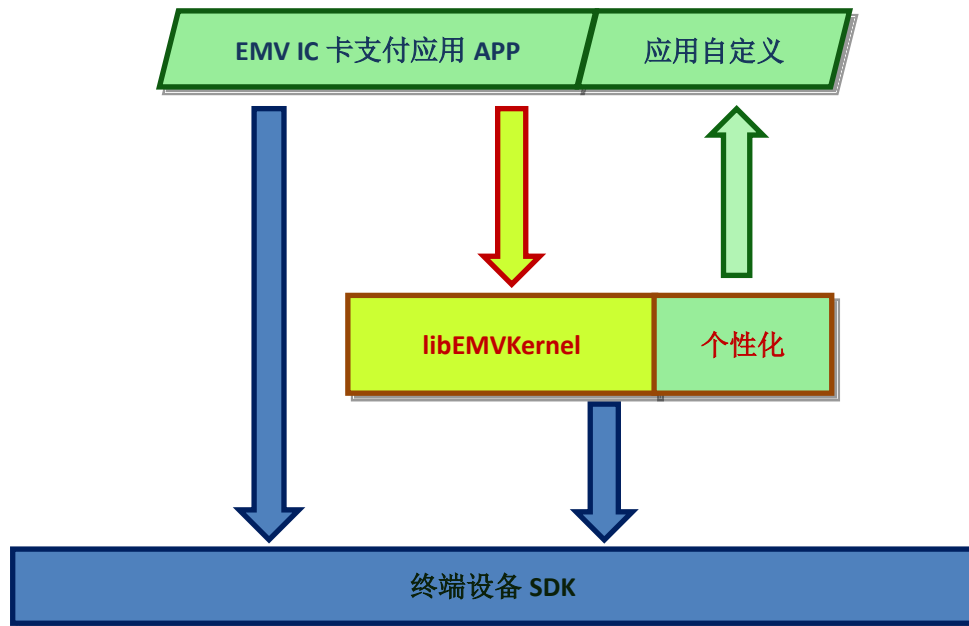
第 8 部分获取内核 Debug 日志，介绍了获取 EMV 内核的 debug 日志的方法和步骤。

第 9 部分附录部分，包含了标签定义列表，宏定义集合，交易返回码，常见错误解答，组成了应用开发过程不可缺少的参考字典。

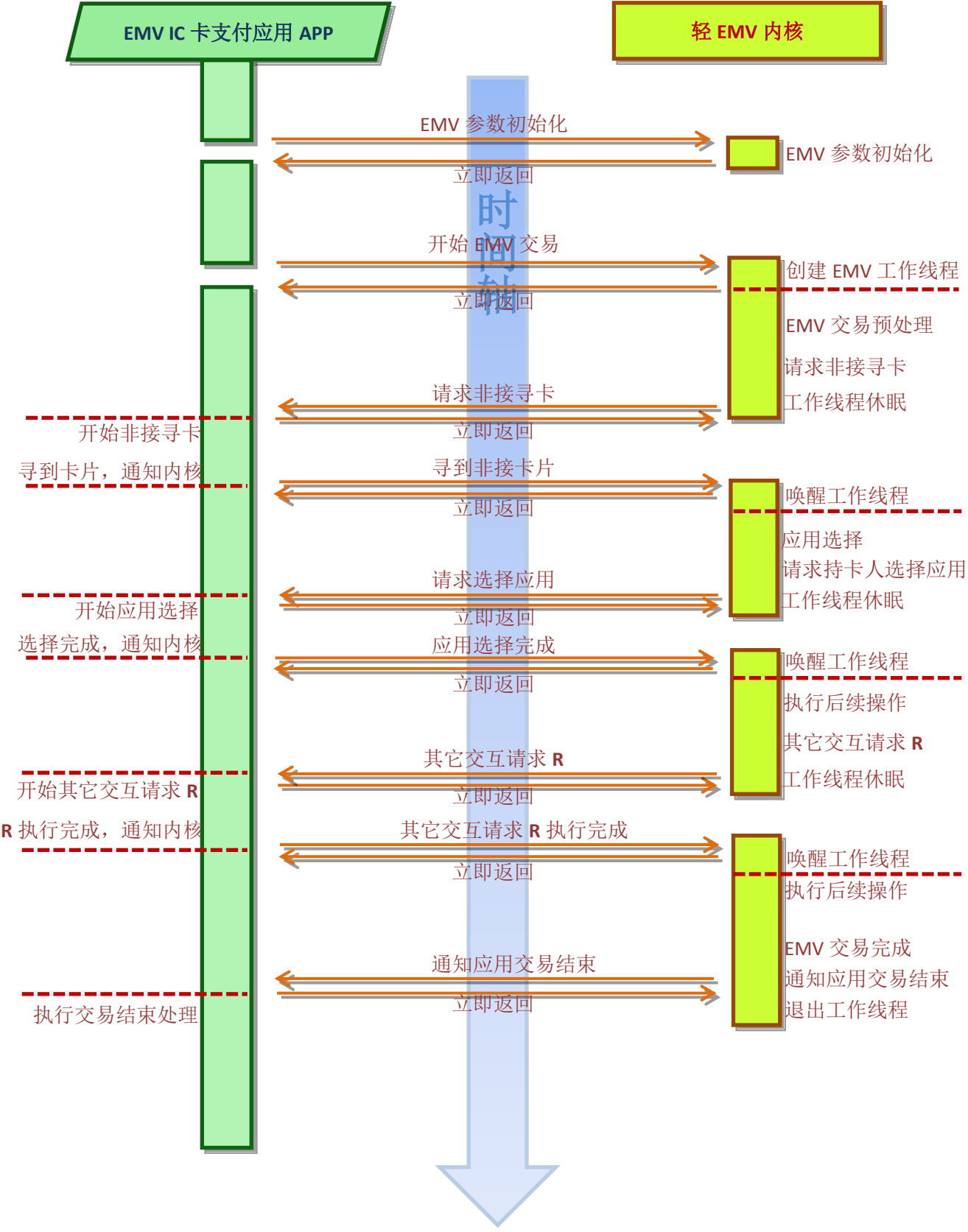
## 2. 模块层次示意图

SDK API 提供终端底层的，与交易无关的，终端各种集成设备的访问接口。

libEMVKernel 是在 SDK 基础上进行封装，集成了 EMV 各类交易流程，并通过 EMV、PBOC、VISA、MASTER、AMEX3.1、DISCOVER 等卡组织的认证的标准库。libEMVKernel 通过开放出简单易用的开发接口，并且通过回调窗口的设计，方便了应用个性化的实现，使之成为一个即灵活便捷又安全稳定的 EMV 应用开发库。



### 3. 交易时序图





## 4. 事件与信号

表格 4-1 列出了内核在 EMV 交易过程中可能产生的事件以及处理事件的信号和数据。

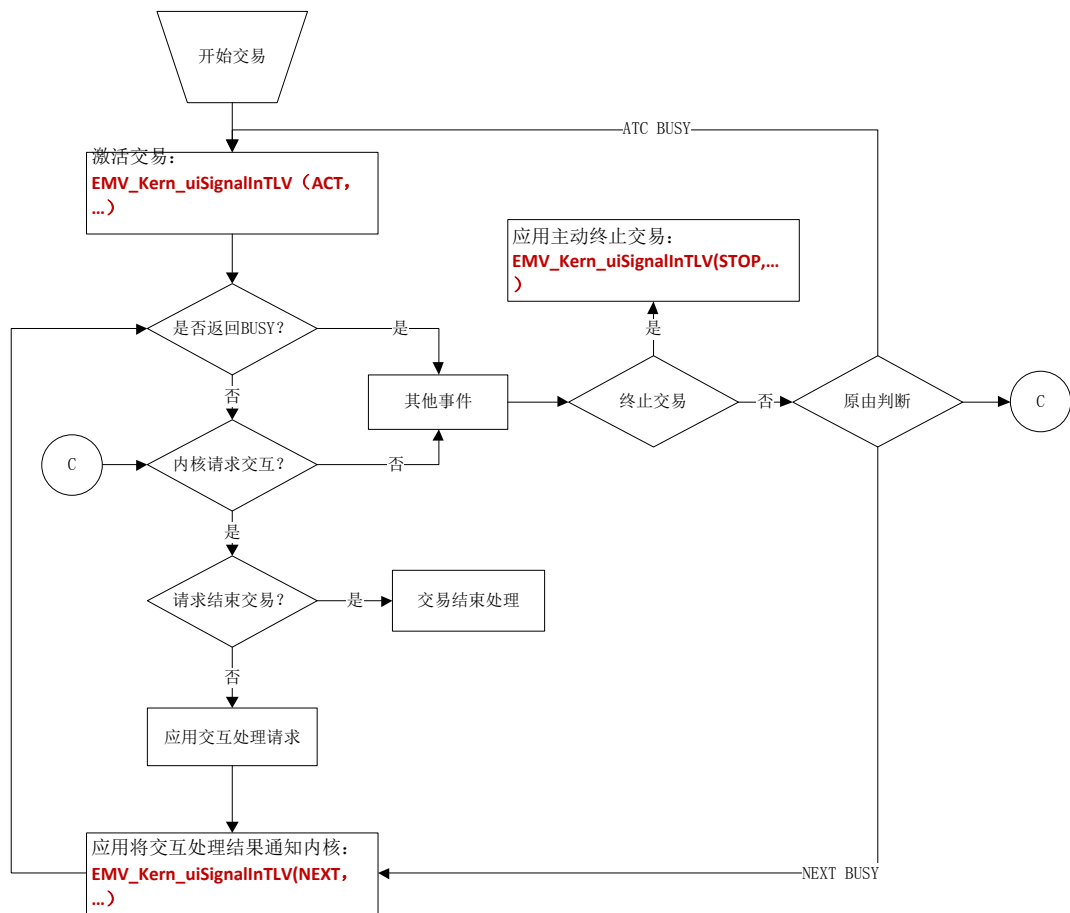
**Table 4-1 事件与信号**

\*备注: (M)-必备 (O)-可选 (C)-条件

事件与通知	回调函数	事件请求描述	等待处理信号	信号数据
		从应用选择开始激活一笔 EMV 交易。	EMV_SIGNAL_ACT	DEF_TAG_PSE_FLAG (O) DEF_TAG_QUERY_ICCLOG (O) DEF_TAG_START_RECOVERY (O)
寻卡事件	EXEP_ucWaitCard	请求持卡人放置卡片	EMV_SIGNAL_NEXT	NULL
应用选择事件	EXEP_ucAppSelection	请求持卡人选择应用	EMV_SIGNAL_NEXT	EMV_TAG_TM_AID (M)
最终选择后事件	EXEP_ucFinalSlt	提供窗口给应用端设置参数，可根据最终选择的 AID 进行参数设置或者调整。	EMV_SIGNAL_NEXT	请参考本文档第 5 部分交易流程图中各类交易流程的参数请求表
读记录后事件	EXEP_ucReadRecord	提供窗口给应用端处理卡片记录数据以及设置参数，例如显示卡号，查找黑名单，设置公钥等。	EMV_SIGNAL_NEXT	DEF_TAG_PAN_IN_BLACK (O) DEF_TAG_ACCUMULATE_AMOUNT (O)
持卡人验证事件	EXEP_ucCardHolderVerify	请求持卡人执行内核指定的持卡人验证。	EMV_SIGNAL_NEXT	DEF_TAG_CHV_STATUS (M)
联机处理事件	EXEP_ucOnlineProcess	请求应用端执行联机授权。	EMV_SIGNAL_NEXT	DEF_TAG_ONLINE_STATUS (M) If online communication is success, following is necessary while returned by host service. EMV_TAG_TM_ARC (C) DEF_TAG_AUTHORIZE_FLAG (C) EMV_TAG_TM_AUTHCODE (C) DEF_TAG_HOST_TLVDATA (C)
交易结束通知	EXEP_vEndProcess	通知应用端 EMV 交易处理完成，内核退出。	无需响应	
数据请求事件	EXEP_vObtain	请求应用端数据，例如闪卡记录。	EMV_SIGNAL_NEXT	根据请求数据类型，传入相应数据。例如，请求闪卡记录，则传入闪卡记录数据。
数据发送通知	EXEP_vSendOut	发送数据给应用端，例如：提示信息，闪卡记录数据，日志数据等。	无需响应	
		在交易处理过程中，请求内核终止交易。	EMV_SIGNAL_STOP	

## 5. 交易流程图

EMV 交易采用事件和信号的机制来实现所有的交易流程。应用激活一笔 EMV 交易后，EMV 内核会主导这笔交易的流程，在需要应用交互行为时，或者需要应用设置交易参数时，内核会向应用发送事件请求，并且等待应用处理事件的结果通知。一旦内核向应用请求了事件，内核将进入休眠，只有应用将事件处理结果通知内核后，内核才会被唤醒并继续执行交易流程，直到内核发出交易结束的通知。

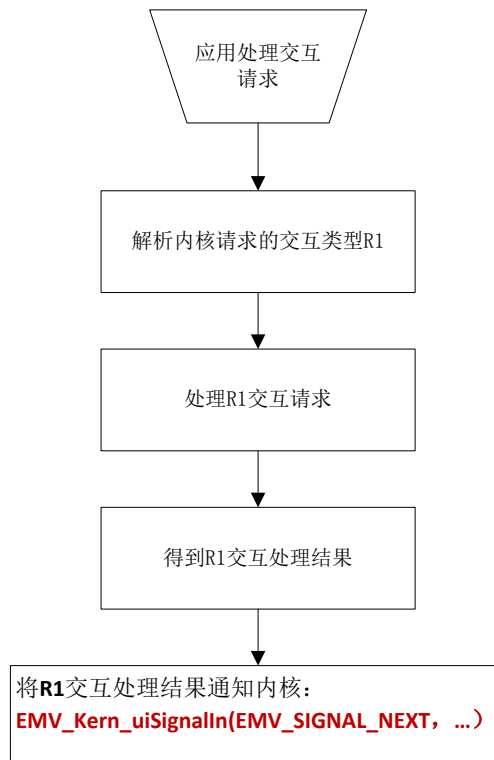


例如：EMV内核回调请求R1， R1的回调函数内部实现如下：

- 1、保存内核反馈给应用的参数
- 2、标记内核请求的交互类型R1
- 3、结束处理，返回成功

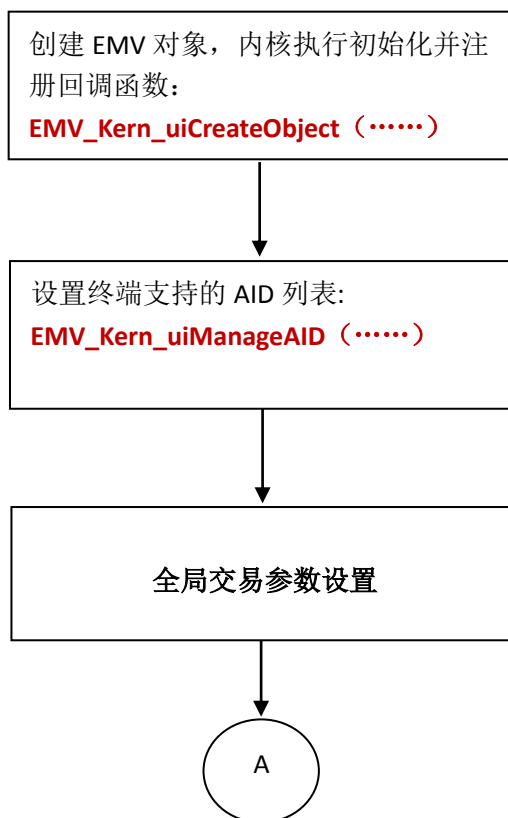
**注意：**永远不要在回调函数内部执行**EMV\_Kern\_uiSignalIn(NEXT)**

参考回调函数实现的示例说明：EMV\_Kern\_uiRegisterIFC

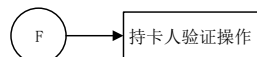
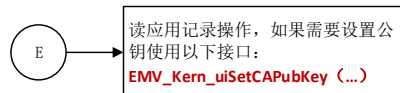
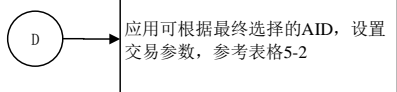
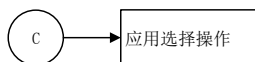
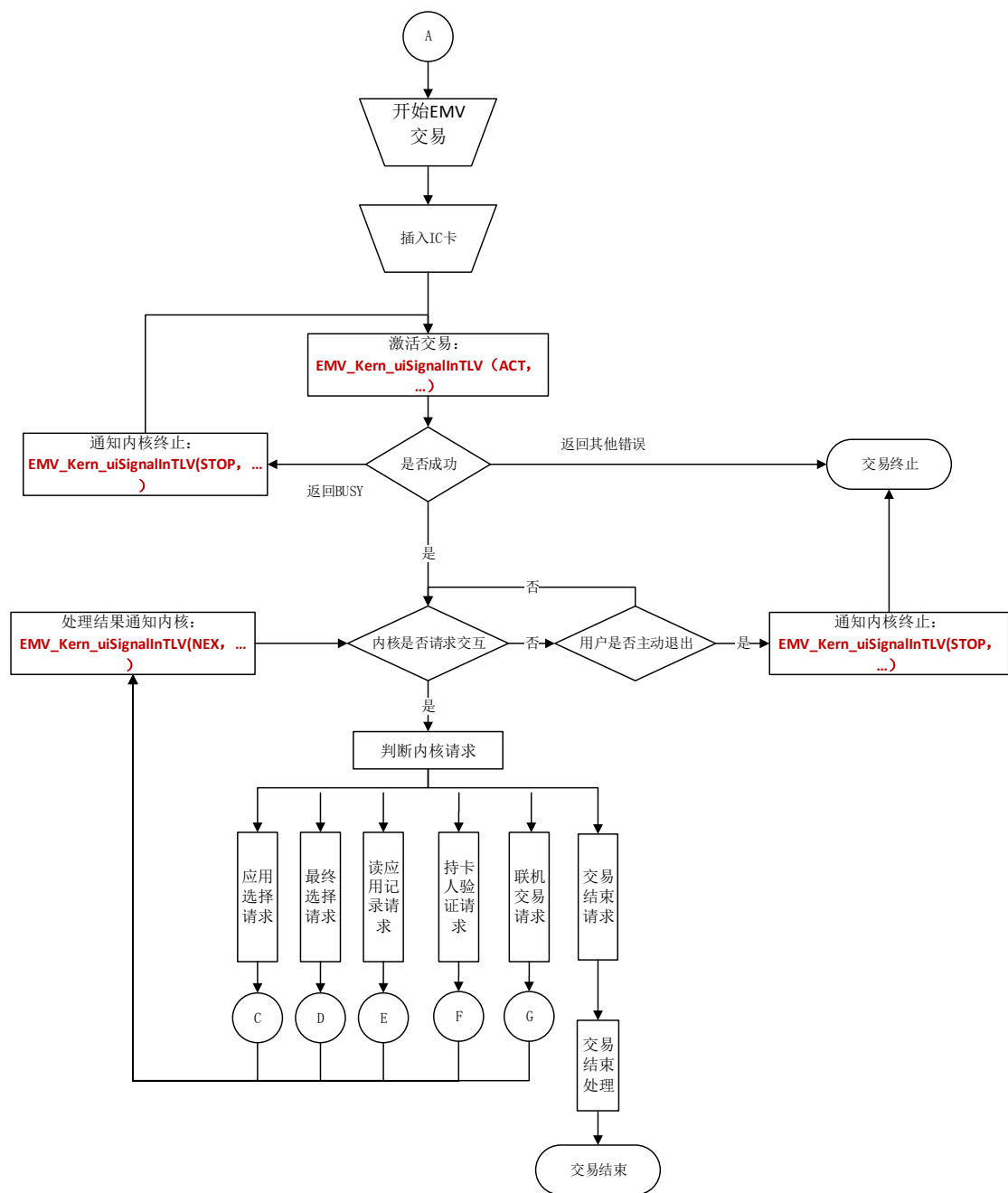


## 5.1 交易初始化

在开始任何交易流程之前，需要对 EMV 内核执行初始化操作。该初始化操作只需要在每次应用开启时执行一次即可。如果应用需要设置，全局的，不因交易类型改变的参数，应该在执行完成初始化后，通过相应的 API 设置到内核。



## 5.2 EMV 标准借贷记交易

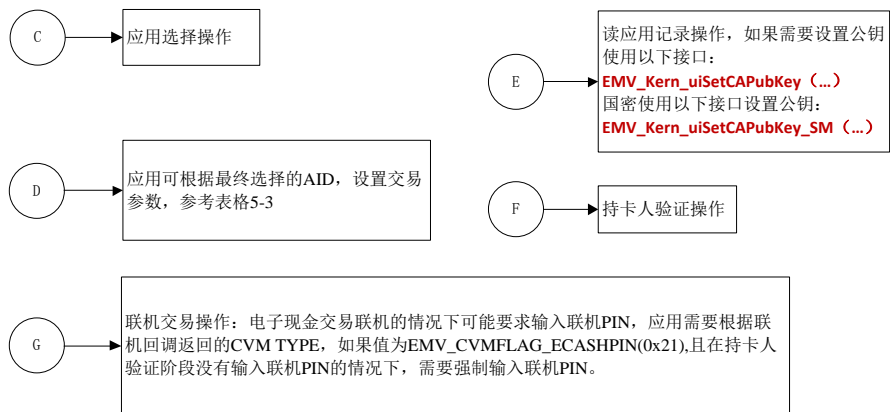
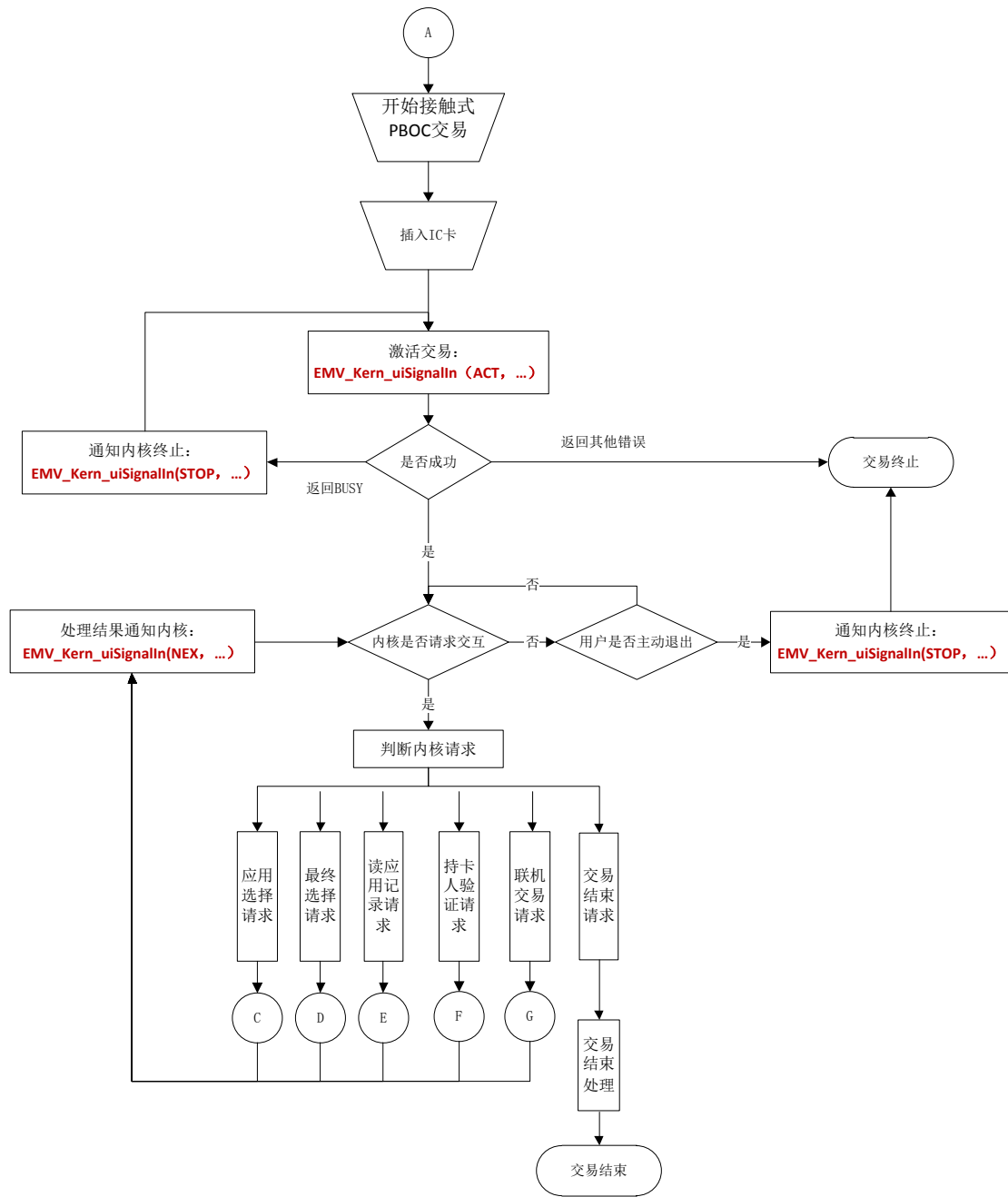


表格 5-2 EMV 标准借贷记交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
D: 最终选择后处理	<p>应用最终选择后，AID 已经确定，应用可根据 AID 来设置不同的交易参数，下面给出本流程需要设置的参数标签：</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><b>TAG 的定义</b>参见<a href="#">附录 A 标签定义</a></p> <p><b>设置接口</b>如下：</p> <pre>EMV_Kern_uiSetTLV(ucKernelID = EMV_KERNELID_EMV) EMV_Kern_uiSetTLVList(ucKernelID = EMV_KERNELID_EMV)</pre>

### 5.3 PBOC 接触式电子现金交易



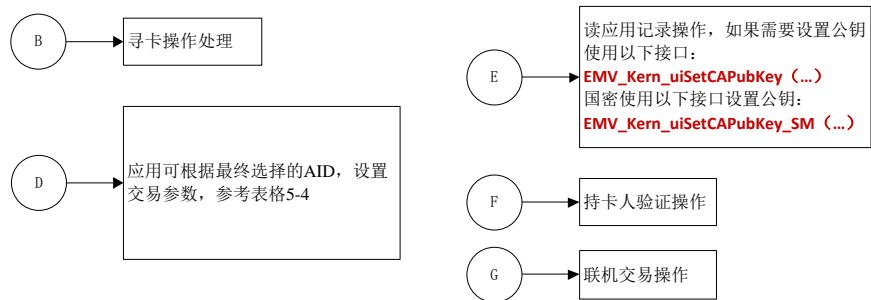
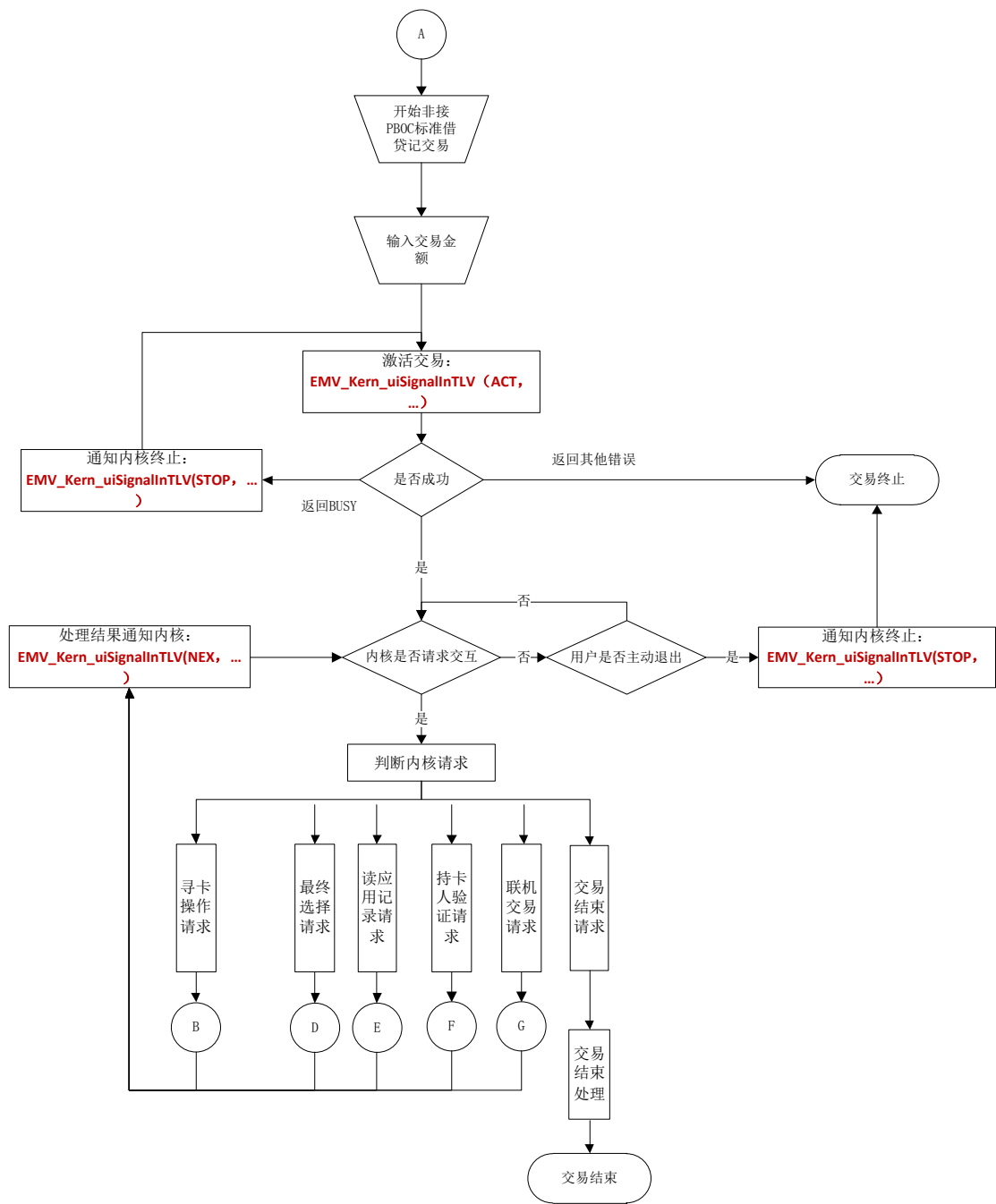
表格 5-3 PBOC 接触式电子现金交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
D: 最终选择后处理	<p>应用最终选择后，AID 已经确定，应用可根据 AID 来设置不同的交易参数，下面给出本流程需要设置的参数标签：</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 的定义参见<a href="#">附录 A 标签定义</a></p> <p>设置接口如下：</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 非接标准借贷记交易

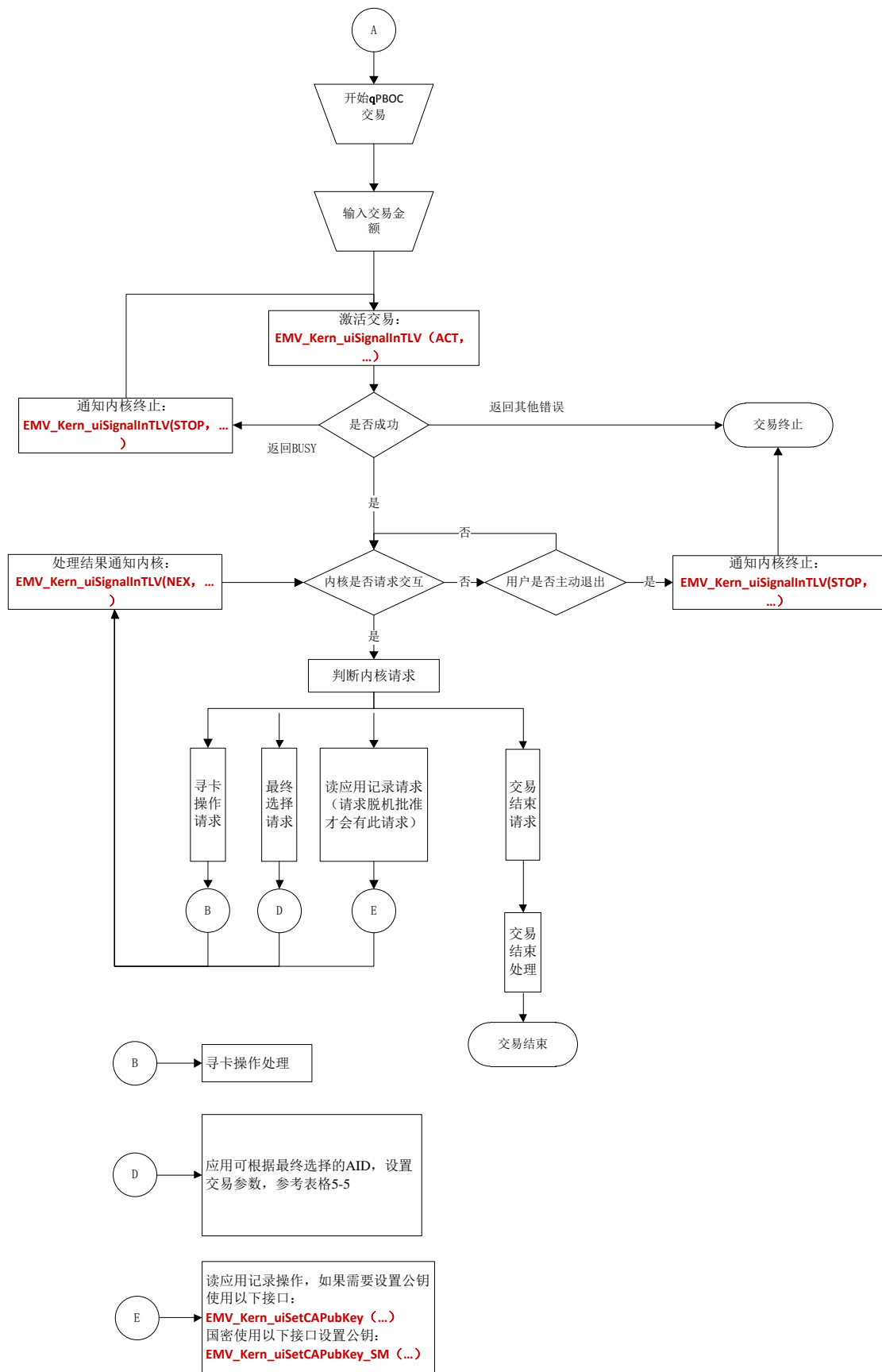


表格 5-4 PBOC 非接标准借贷记交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
D: 最终选择后处理	<p>应用最终选择后, AID 已经确定, 应用可根据 AID 来设置不同的交易参数, 下面给出本流程需要设置的参数标签:</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: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>• C_TAG_TM_RD_RCP (O)REF_V:0x7C00</li></ul> <p><b>TAG</b> 的定义参见<a href="#">附录 A 标签定义</a></p> <p>设置接口如下:</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 非接快速支付交易(qPBOC)

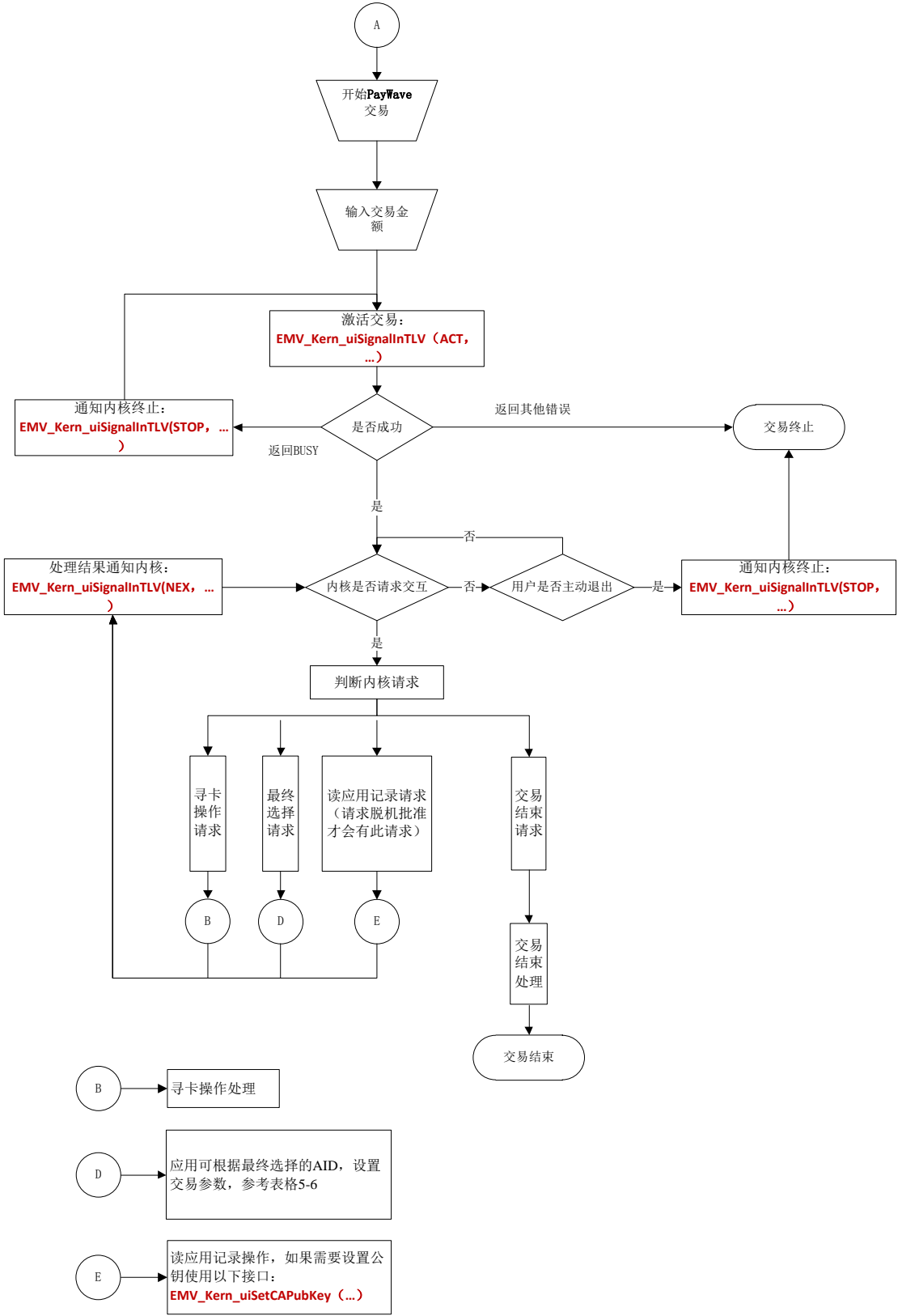


表格 5-5 PBOC 非接快速支付交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
D: 最终选择后处理	<p>应用最终选择后，AID 已经确定，应用可根据 AID 来设置不同的交易参数，下面给出本流程需要设置的参数标签：</p> <ul style="list-style-type: none"><li>• EMV_TAG_TM_AUTHAMNTN (M)REF_V:0x000000000100(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><b>TAG 的定义</b>参见<a href="#">附录 A 标签定义</a></p> <p><b>设置接口</b>如下：</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)交易



表格 5-6 VISA PayWave 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
ACT: ACT 激活时	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03</li> </ul> <p>TAG 的定义参见<a href="#">附录 A 标签定义</a> 设置接口如下:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>) <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>
D: 最终选择后处理	<p>应用最终选择后, AID 已经确定, 以下列出本阶段需要设置的全部参数标签: (在 <a href="#">4.事件与信号</a>中与本阶段相同的 Tag 无需重复设置)</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 的定义参见<a href="#">附录 A 标签定义</a> 设置接口如下:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>) <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>
E: 读完应用数据后	<ul style="list-style-type: none"> <li>DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> </ul> <p>TAG 的定义参见<a href="#">附录 A 标签定义</a> 设置接口如下:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>) <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_VISA</a>)</p>

备注:

<sup>1</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B)是 4 字节 HEX 码, 不是 6 字节, 如 100.00 元设置 “\x00\x00\x27\x10”, 应用设置时需要注意。

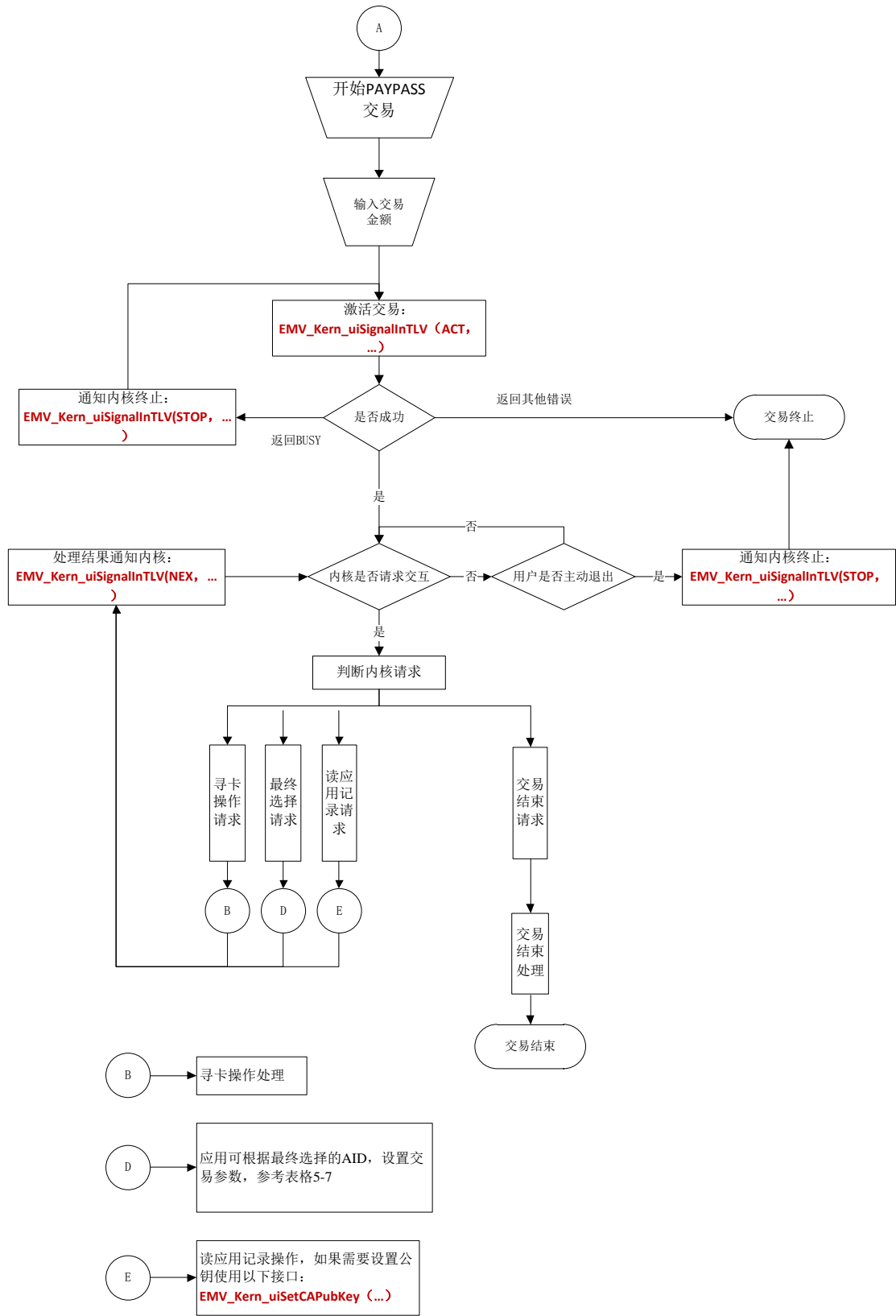
VISA 非接快速支付交易[脱机批准交易](#)，不会发起联机请求事件，而是在交易结束通知事件产生后，由输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作，后续处理无需与内核交互。（注：脱机批准不会请求联机 PIN）

交易结束事件的返回码，可详见《[附录 C 交易返回码](#)》。

以下对《附录 C 的交易返回码》之 EMV\_RESULT\_REPOWERICC 做 Visa 交易的特别说明。

当在 GPO 阶段卡片返回 6986 时内核会返回 EMV\_RESULT\_REPOWERICC，要求重新挥卡。应用端要在交易参数不变(交易金额，交易类型等)的情况下，提示持卡人“Refer to your Device”，下电等待 1000ms-1500ms 后重新上电并激活卡片并寻卡，重新挥卡来完成后续的完整交易。(以上是认证时要求的处理，仅供参考。应用可以根据实际需求做相应处理)

## 5.7 MASTERCARD(PayPass)交易





表格 5-7 MasterCard Paypass 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数			
D: 最终选择后处理	应用最终选择后，AID 已经确定，应用可根据 AID 来设置不同的交易参数，下面给出本流程需要设置的参数标签：			
	参数标签	属性	参照值	缺省值
	EMV_TAG_TM_TERMTYPE	M	0x22	
	EMV_TAG_TM_CAP	O	0xE0F808	0x000000
	EMV_TAG_TM_CAP_AD	O	0x6000F0A001	0x0000000000
	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	0x840000000C
	DEF_TAG_TAC_ONLINE	O	0x840000000C	0x840000000C
	DEF_TAG_TAC_DEFAULT	O	0x840000000C	0x840000000C
	DEF_TAG_M_TRANS_MOD	O	0x02	0x02
	DEF_TAG_M_BALANCE_SUP	O	0x00	0x00
	DEF_TAG_TORN_SUPPORT	O	0x00	0x00
	DEF_TAG_M_CDV_SUP	O	0x00	0x00
	M_TAG_TM_TRANS_LIMIT	M	0x000000030000 (300.00)	0x000000000000
	M_TAG_TM_TRANS_LIMIT_CDV	M	0x000000050000 (500.00)	0x000000000000
	M_TAG_TM_CVM_LIMIT	O	0x000000001000( 10.00)	0x000000000000
	M_TAG_TM_FLOOR_LIMIT	O	0x000000010000( 100.00)	0x000000000000
	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	0x010101010101 01010101010101 01010101010101	
	M_TAG_TM_9F53	O	0x01	
	M_TAG_TM_9F6D	O	0x0001	0x0001
	TAG 的定义参见 <a href="#">附录 A 标签定义</a>			

	设置接口如下:  <b>EMV_Kern_uiSetTLV</b> <b>EMV_Kern_uiSetTLVList</b>
--	---

备注:

1、PayPass 磁条模式获取 Track1 和 Track2 方式有三种:

1)通过 API 接口 EMV\_Kern\_uiGetTLV 获取。

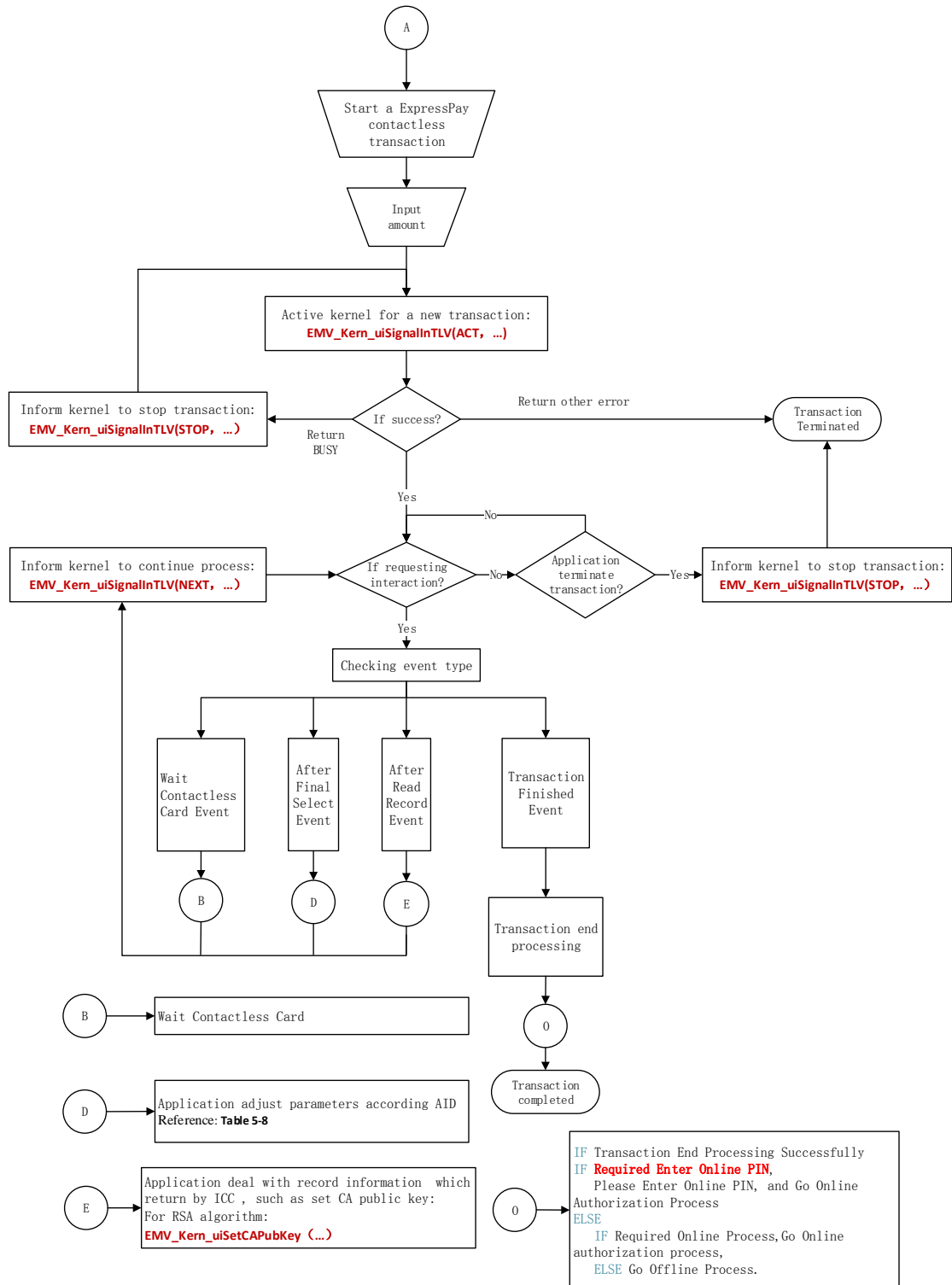
2)通过交易结束的交易记录 EMV\_tTransData 字段 auTLVData 获取需要的标签。

3)通过交易记录 EMV\_tTransData 中字段,

auMSDT1Data/ucMSDT1Len/auMSDT2Data/ucMSDT2Len 获取内容。

4)兼容存量, DEF\_TAG\_M\_TORN\_TRANS 也可设置闪卡功能, 新开发应用建议别用此标签, 若同时设置 DEF\_TAG\_TORN\_SUPPORT 和 DEF\_TAG\_M\_TORN\_TRANS, 则以 DEF\_TAG\_M\_TORN\_TRANS 为主。

## 5.8 AMEX 美国运通交易



表格 5-8 AMEX 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
ACT: ACT 激活时	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03</li> <li><a href="#">DEF_TAG_PPSE_6A82_TURNTO_AIDLIST</a>(M) REF_V:0x01</li> </ul> <p>TAG 的定义参见<a href="#">附录 A 标签定义</a> 设置接口如下:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>) <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p>
D: 最终选择后处理	<p>应用最终选择后, AID 已经确定, 以下列出本阶段需要设置的全部参数标签: (在 <a href="#">4.事件与信号</a> 中与本阶段相同的 Tag 无需重复设置)</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(固定值)</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 href="#">A_TAG_PREAGAIN</a>(C) REF_V:0x00 or 0x01</li> </ul> <p>TAG 的定义参见<a href="#">附录 A 标签定义</a> 设置接口如下:</p> <p><b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>) <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_AMEX</a>)</p>
E: 读完应用数据后	<ul style="list-style-type: none"> <li><a href="#">DEF_TAG_PAN_IN_BLACK</a>(O) REF_V:0x00 or 0x01</li> </ul>

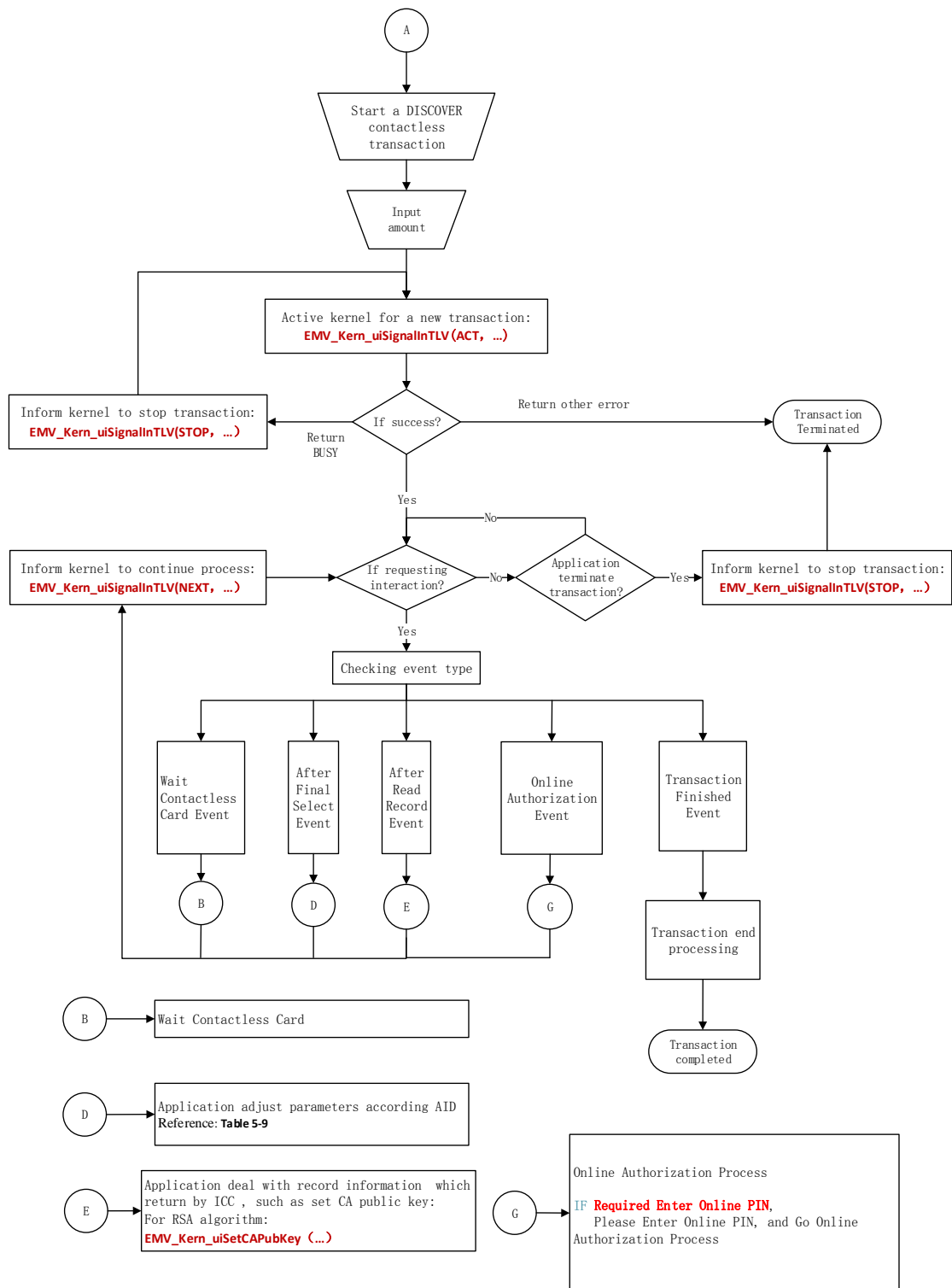
<sup>2</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B)是 4 字节 HEX 码, 不是 6 字节, 如 100.00 元设置 “\x00\x00\x27\x10”, 应用设置时需要注意。

	<ul style="list-style-type: none"><li>• A_TAG_TM_TRANS_LIMIT(C) REF_V:<a href="#">Sets of Dynamic Reader Limits</a></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 href="#">A_TAG_TM_IN_CARD_BIN_RANGE</a>(O) REF_V:0xA0 or 0x00</li></ul> <p><b>TAG</b> 的定义参见<a href="#">附录 A 标签定义</a></p> <p>设置接口如下:</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>
--	--

备注:

Amex 非接快速支付交易[脱机批准交易](#)，不会发起联机请求事件，而是在交易结束通知事件产生后，由输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作，后续处理无需与内核交互。

## 5.9 DISCOVER 大莱交易



表格 5-9 DISCOVER 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
<b>ACT: ACT 激活时</b>	<ul style="list-style-type: none"> <li>DEF_TAG_PSE_FLAG(M) REF_V:0x03 (ZIP Mode 参考 <a href="#">6.3 EMV Kern uiManageAID</a> ucPartSlt 的 b2 位)</li> <li>EMV_TAG_TM_AUTHAMNTN(M) REF_V:0x00000000100(1.00)</li> <li>EMV_TAG_TM_OTHERAMNTN(O) REF_V:0x0000000000</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><a href="#">DEF_TAG_D_ISSUERSCRIPT_EXCUTIVE</a>(C) REF_V:0x01 or 0x00</li> </ul> <p><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b>  <b>设置接口如下:</b>  <b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)  <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>
<b>D: 最终选择后处理</b>	<p>应用最终选择后, AID 已经确定, 以下列出本阶段需要设置的全部参数标签: (在 <a href="#">4.事件与信号</a> 中与本阶段相同的 Tag 无需重复设置)</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:0x0000000000</li> <li>DEF_TAG_TAC_ONLINE(O) REF_V:0x0000000000</li> <li>DEF_TAG_TAC_DEFAULT(O) REF_V:0x0000000000</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:0x00000001</li> </ul> <p><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b>  <b>设置接口如下:</b>  <b>EMV_Kern_uiSetTLV</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)  <b>EMV_Kern_uiSetTLVList</b>(ucKernelID = <a href="#">EMV_KERNELID_DISCOVER</a>)</p>

<sup>3</sup>EMV\_TAG\_TM\_FLOORLMT(Tag: 9F1B)是 4 字节 HEX 码, 不是 6 字节, 如 100.00 元设置 “\x00\x00\x27\x10”, 应用设置时需要注意。

E: 读完应用数据后	<ul style="list-style-type: none"> <li>DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> </ul> <p><b>TAG 的定义</b>参见<a href="#">附录 A 标签定义</a></p> <p><b>设置接口</b>如下:</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>
G:发起联机处理后	<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><b>TAG 的定义</b>参见<a href="#">附录 A 标签定义</a></p> <p><b>设置接口</b>如下:</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>

备注:

Discover 非接快速支付交易**脱机批准交易**，不会发起联机请求事件，而是在交易结束通知事件产生后，由输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作，后续处理无需与内核交互。

而 Discover 非接快速交易的**联机交易**，会请求联机请求事件，且由联机请求的输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作，继续执行联机事件处理，直至交易结束事件。此时的交易结束事件，内核不会再次请求持卡人验证，即一个交易内核只会请求一次持卡人验证。

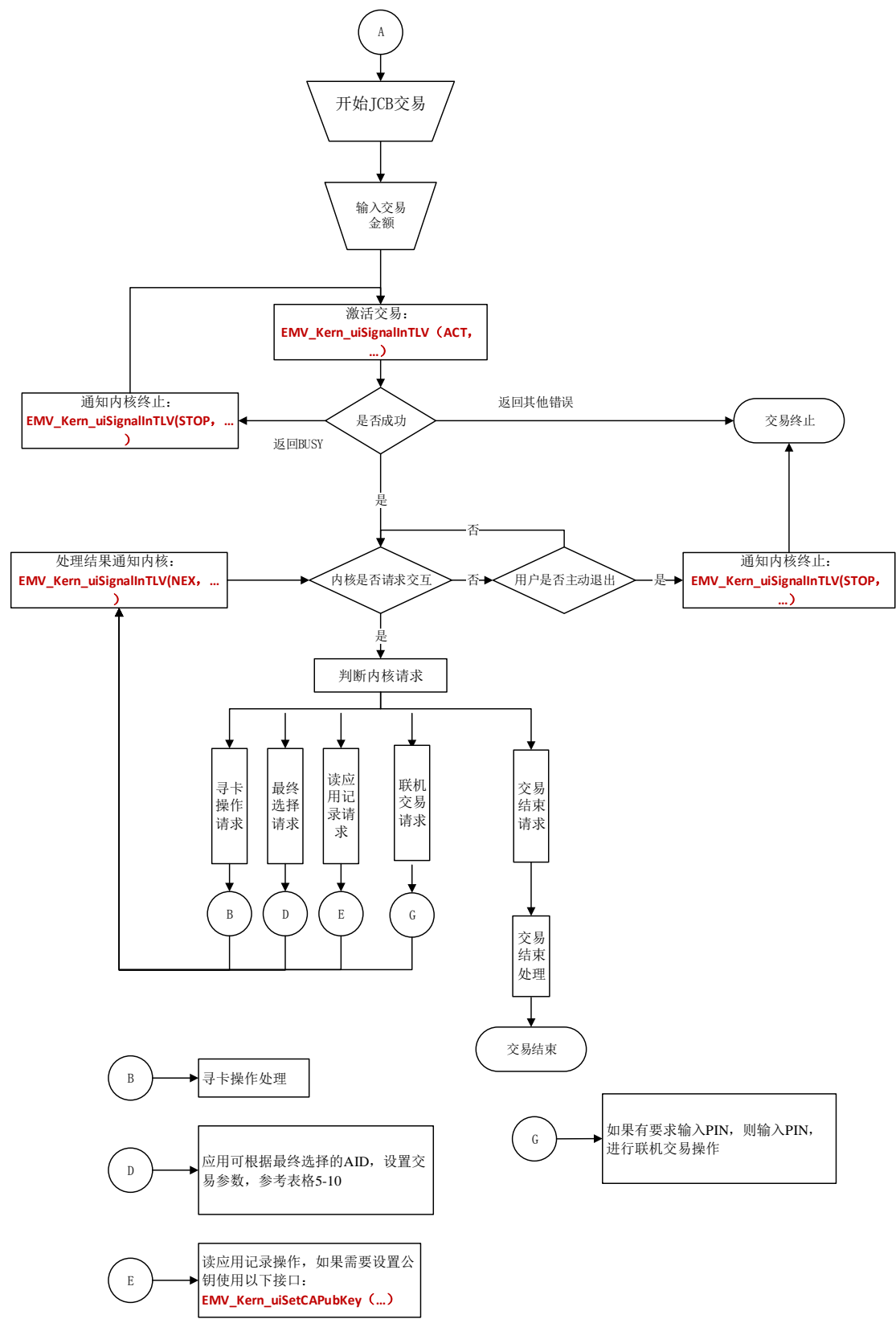
关于发卡行脚本处理:

在联机请求事件，应用端发起联机请求，Host 如果有下发 Tag71 或 Tag72 脚本，需要做发卡行脚本更新处理的。应用端应进行如下步骤处理：应用端要在交易参数不变(交易金额，交易类型等)的情况下，无需关闭射频模块，直接在 ACT 阶段设置

DEF\_TAG\_D\_ISSUERSCRIPT\_EXCUTIVE 值为 0x01 给内核，内核会接着请求 OnWaitCard(Flag=2) 事件,应用端接着第二次出卡寻卡并激活卡片,内核接着执行 SELECT AID 后,执行 Issuer Script Update 脚本。脚本执行完成后，内核会在交易结束事件中的 ACType 请求 ARQC,后续应用端可以根据实际需求做响应处理，如马上发起脚本结果通知，或不马上发起脚本结果通知等。



5.10 JCB 日财卡交易

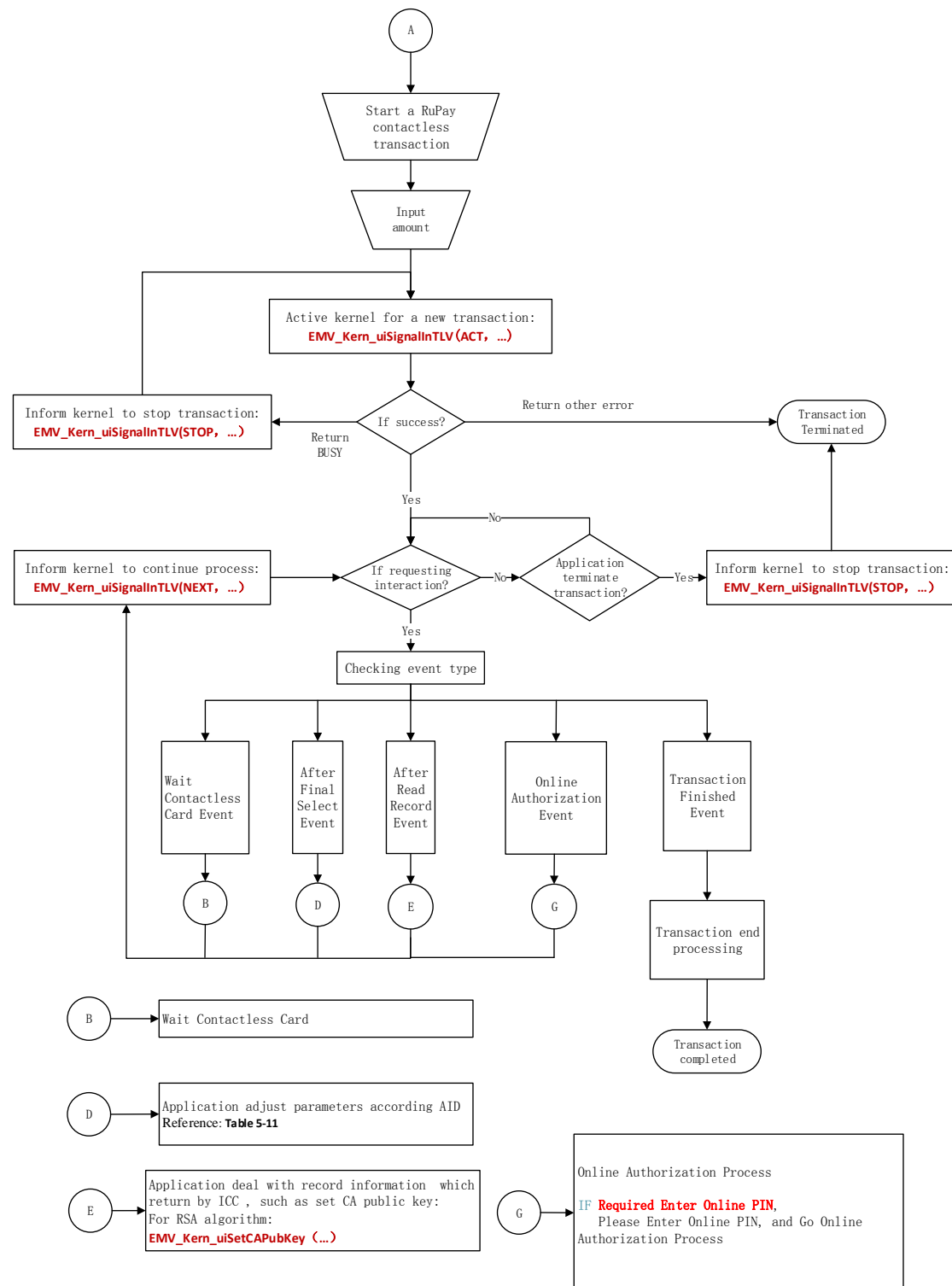


表格 5-10 JCB 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件

事件类型	请求交易参数
D: 最终选择后处理	<p>应用最终选择后, AID 已经确定, 应用可根据 AID 来设置不同的交易参数, 下面给出本流程需要设置的参数标签:</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_AUTHAMNTN(M) REF_V(6):0x000000000100(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):0x000000000000</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):0x0000000000010</li> <li>• EMV_TAG_TM_MCHCATCODE(O) REF_V(2):0x7032</li> <li>• EMV_TAG_TM_MCHNAMELOC(O) REF_V(0_32):0x5858204D45524348414E54205959204C4F43 4154494F4E(23byte)</li> <li>• DEF_TAG_J_TRANS_LIMIT(O) REF_V(6):0x0000000020000(200.00)</li> <li>• DEF_TAG_J_CVM_LIMIT(O) REF_V(6):0x0000000010000(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><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b></p> <p><b>设置接口如下:</b></p> <p><b>EMV_Kern_uiSetTLV</b></p> <p><b>EMV_Kern_uiSetTLVList</b></p>

## 5.11 RUPAY(qSPARC)



表格 5-11 RUPAY 交易参数请求

REF\_V-参考值 M-必备 O-可选 C-条件 R-Recommend

事件类型	请求交易参数																																					
ACT: ACT 激活时	<div><div><div><div><div>•</div><div>DEF_TAG_PSE_FLAG(M)</div><div>REF_V:0x03</div></div><div>•</div><div>EMV_TAG_TM_AUTHAMNTN(M)</div><div>REF_V:0x000000000100(1.00)</div></div><div>•</div><div>EMV_TAG_TM_OTHERAMNTN(O)</div><div>REF_V:0x0000000000</div></div><div>•</div><div>R_TAG_TM_TRANS_LIMIT(M)</div><div>REF_V:0x000000015000</div></div> <div>•</div> <div>EMV_TAG_TM_CAP(M)</div> <div>REF_V:0xE06840</div> <div><div>1)</div><div>About CVM capability: RuPay only support 3 types of CVM, show bellow:</div><div>Online PIN B2b7=1,</div><div>Signature B2b6=1,</div><div>NoCVM B2b4=1.</div></div> <div><div>2)</div><div>About security capability: RuPay only support qDDA for contactless, B3b7=1,</div><div>so Byte3=0x40 is recommend.</div></div> <div><div>•</div><div>DEF_TAG_START_RECOVERY(C)</div><div>REF_V: 0x00 or 0x01 valid for RuPay</div></div> <div><div>•</div><div>R_TAG_TM_TIMELIMIT(O)</div><div>REF_V:0x1E(30 seconds)</div></div> <div>TAG 的定义参见<a href="#">附录 A 标签定义</a></div> <div>设置接口如下:</div> <div><div>EMV_Kern_uiSetTLV(ucKernelID = EMV_KERNELID_RUPAY)</div><div>EMV_Kern_uiSetTLVList(ucKernelID = EMV_KERNELID_RUPAY)</div></div>																																					
D: 最终选择后处理	<div>应用最终选择后, AID 已经确定, 以下列出本阶段需要设置的全部参数标签: (在 <a href="#">4.事件与信号</a>中与本阶段相同的 Tag 无需重复设置)</div> <div><div><div>•</div><div>EMV_TAG_TM_TERMTYPE(M)</div><div>REF_V: show bellow</div></div><table><tr><th>Operating Filed</th><th>Terminal Type</th><th>Value</th><th>Use case</th></tr><tr><td rowspan="5">Retail</td><td>Attended Online only</td><td>21</td><td>Merchants</td></tr><tr><td>Attended Offline with Online capability</td><td>22</td><td>Merchants</td></tr><tr><td>Attended Offline only</td><td>23</td><td>Merchants</td></tr><tr><td>Unattended Offline with Online capability</td><td>25</td><td>Vending Machines</td></tr><tr><td>Unattended Offline only</td><td>26</td><td>Vending Machines</td></tr><tr><td rowspan="6">Transit</td><td rowspan="2">Attended Offline with Online capability</td><td>92</td><td>Transit with conductor</td></tr><tr><td>93</td><td>Fare Adjustment Terminal</td></tr><tr><td>Attended Offline only</td><td>94</td><td>Transit with conductor</td></tr><tr><td rowspan="2">Unattended Offline with Online capability</td><td>96</td><td>Transit without conductor</td></tr><tr><td>97</td><td>Fare Adjustment Terminal</td></tr><tr><td>Unattended Offline only</td><td>98</td><td>Transit without conductor</td></tr></table></div> <div><div>•</div><div>EMV_TAG_TM_CAP_AD(M)</div><div>REF_V:0xF040F0B001</div><div>For this specification, Tag 9F40 B2b7 shall be 1 indicates service creation is support for the terminal to backward compatible with Legacy card.</div></div> <div><div>•</div><div>R_TAG_TM_SRTRANS_CAP(C)</div><div>REF_V:0x002C</div><div><div>1)</div><div>B2b6: Transit Transaction</div></div></div>	Operating Filed	Terminal Type	Value	Use case	Retail	Attended Online only	21	Merchants	Attended Offline with Online capability	22	Merchants	Attended Offline only	23	Merchants	Unattended Offline with Online capability	25	Vending Machines	Unattended Offline only	26	Vending Machines	Transit	Attended Offline with Online capability	92	Transit with conductor	93	Fare Adjustment Terminal	Attended Offline only	94	Transit with conductor	Unattended Offline with Online capability	96	Transit without conductor	97	Fare Adjustment Terminal	Unattended Offline only	98	Transit without conductor
Operating Filed	Terminal Type	Value	Use case																																			
Retail	Attended Online only	21	Merchants																																			
	Attended Offline with Online capability	22	Merchants																																			
	Attended Offline only	23	Merchants																																			
	Unattended Offline with Online capability	25	Vending Machines																																			
	Unattended Offline only	26	Vending Machines																																			
Transit	Attended Offline with Online capability	92	Transit with conductor																																			
		93	Fare Adjustment Terminal																																			
	Attended Offline only	94	Transit with conductor																																			
	Unattended Offline with Online capability	96	Transit without conductor																																			
		97	Fare Adjustment Terminal																																			
	Unattended Offline only	98	Transit without conductor																																			

	<p>2) B2b5: CAT3 Transaction</p> <p>3) B2b4: Money Add Transaction</p> <p>4) B2b3: Void Transaction</p> <p>5) Other bits was RFU</p> <ul style="list-style-type: none"> <li>• EMV_TAG_TM_CNTRYCODE(M) REF_V:0x0356</li> <li>• EMV_TAG_TM_CURCODE(M) REF_V:0x0356</li> <li>• EMV_TAG_TM_APPVERNO(O) REF_V:0x0002</li> </ul> <p>For this specification for this tag is a fixed value, kernel will be 0x0002 default value if application not setting.</p> <ul style="list-style-type: none"> <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>• 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:0x00000001</li> <li>• R_TAG_TM_DF16(C) REF_V:0x0000</li> </ul> <p><i>If card or terminal not found mutual support Service ID or terminal not support service-based transaction, the value always shall be all zeros.</i></p> <p><i>If card and terminal support service transaction and also found mutual support Service ID, Please refer to Section 5.11.1 for more detail.</i></p> <ul style="list-style-type: none"> <li>• R_TAG_TM_DF3A(O) REF_V:0040000000</li> </ul> <p><i>If terminal support Services Creation Processing Capabilities, Tag DF3A B2b7 shall be to 1.</i></p> <p><i>If terminal not support Service Creation, the value shall be set all zeros.</i></p> <ul style="list-style-type: none"> <li>• R_TAG_TM_CVM_LIMIT(R) REF_V:0x000000015000</li> <li>• R_TAG_TM_FLOOR_LIMIT(O)</li> <li>• EMV_TAG_TM_FLOORLMT(O)</li> </ul> <p><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b></p> <p><b>设置接口如下:</b></p> <p><b>EMV_Kern_uiSetTLV(ucKernelID = EMV_KERNELID_RUPAY)</b></p> <p><b>EMV_Kern_uiSetTLVList(ucKernelID = EMV_KERNELID_RUPAY)</b></p>
E: 读完应用数据后	<ul style="list-style-type: none"> <li>• DEF_TAG_PAN_IN_BLACK(O) REF_V:0x00 or 0x01</li> <li>• DEF_TAG_RAND_SLT_THRESHOLD (M) REF_V:0x000000005000(50.00)</li> <li>• DEF_TAG_RAND_SLT_PER (M) REF_V:0x30(BCD)</li> <li>• DEF_TAG_RAND_SLT_MAXPER (M) REF_V:0x90(BCD)</li> <li>• R_TAG_TM_TSRQ(C) REF_V:0x08 1010 9500</li> <li>• R_TAG_TM_DF45(C) REF_V:Var. Max length &lt;=96</li> <li>• R_TAG_IC_DF47(C)(PRMiss For Legacy Mode or For Non-Legacy PRMacq key versioning)</li> <li>• R_TAG_TM_DF48(C)(PRMacq For Key plant Only)</li> <li>• R_TAG_IC_KCV(C)(PRMacq KCV For Key plant Only)</li> <li>• R_TAG_IC_PRMACQ_KEYINDEX(C)(PRMacq Index For Key plant Only)</li> </ul> <p><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b></p>

	设置接口如下: <b>EMV_Kern_uiSetTLV(ucKernelID = EMV_KERNELID_RUPAY)</b> <b>EMV_Kern_uiSetTLVList(ucKernelID = EMV_KERNELID_RUPAY)</b>
<b>G:发起联机处理后</b>	<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> <li>• R_TAG_TM_ONLINE_ORNOT(C) REF_V:0x00 or 0x01</li> </ul> <p><b>TAG 的定义参见<a href="#">附录 A 标签定义</a></b></p> <p>设置接口如下:  <b>EMV_Kern_uiSetTLV(ucKernelID = EMV_KERNELID_RUPAY)</b>  <b>EMV_Kern_uiSetTLVList(ucKernelID = EMV_KERNELID_RUPAY)</b></p>

备注:

对于 RuPay Non-Legacy Mode, 如果走部分联机流程, 不会发起联机请求事件, 而是在交易结束通知事件产生后, 由输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作, 后续处理无需与内核交互。

对于 RuPay Non-Legacy Mode or Legacy Mode, 如果走完整联机流程, 会请求联机请求事件, 且由联机请求的输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作, 继续执行联机事件处理, 直至交易结束事件。此时的交易结束事件, 内核不会再次请求持卡人验证, 即一个交易内核只会请求一次持卡人验证。

### 5.11.1 About Service IDs

Terminal always maintain a list of Service IDs from Acquirer if both terminal and Acquirer support Service-based transaction.

Card will response a Service Directory(Tag DF07) which contain a list of Service IDs that card supported, and in Final Selection APDU card will provide Services IDs if the card support a Service-based transaction.

In Final Selection callback Event, the application shall setting Service ID(DF16) to kernel if found a same Service ID that they support mutually and the Service ID must be non-zero value.

Tag DF07 defined, please refer to **R\_TAG\_IC\_DF07** in table <**A.9 TAG of EMV\_KERNELID\_RUPAY**>.

For example,

a)Found a match Service ID

Terminal maintain a Service IDs:

Service ID, Service qualifier, Service data

1010, 08 1010 9500, 09 10 1506 1501 01 1122334455667788 000001 0001 0061 150406125703 000000010000 02 00 00 000A0101 000A0102 05 02000102010206000A01010302

Card Response Service IDs:

DF07=0100607484123000000020010F000A000000000000000001011AC0010109500

They found a same Service ID('1010') support mutually, so SetTLV(KID=RUPAY, '1010') to kernel.

b)Not Found a match Service ID

Terminal maintain a Service IDs:

Service ID, Service qualifier, Service data



### 5.11.3 About 2-Tap

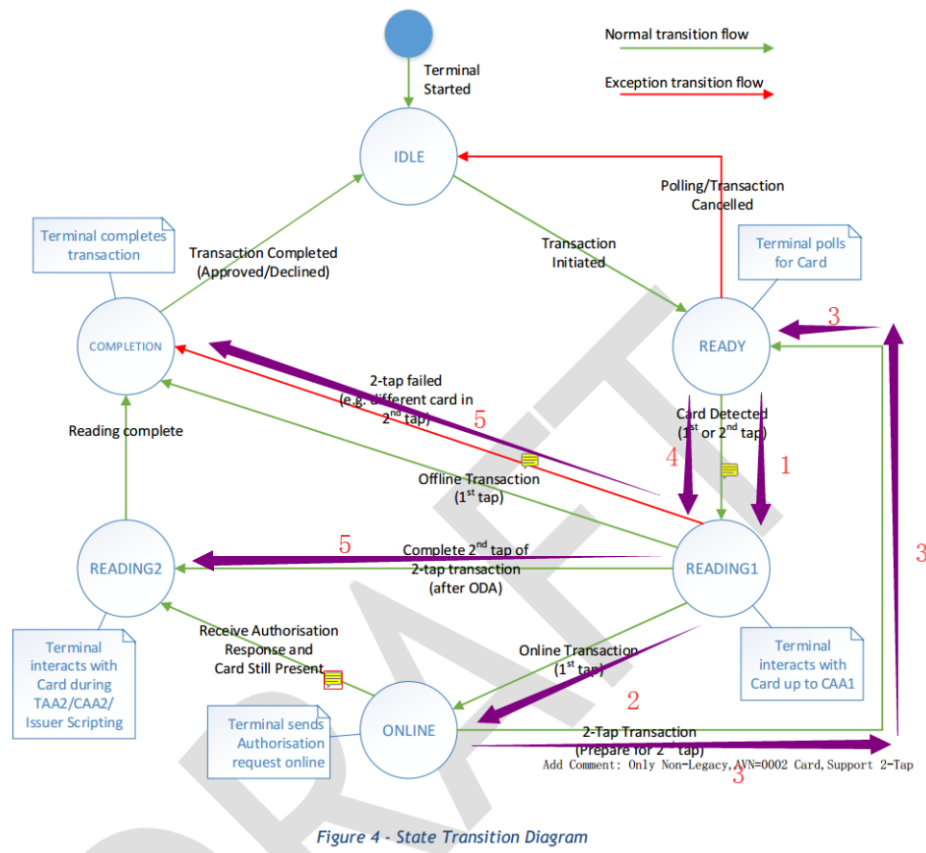


Figure 4 - State Transition Diagram

According to above program (Arrow 1->2->3->4->5), we can know that.

After **ONLINE** processing (Arrow 2), if the Card supports 2-Tap (i.e. is a non-legacy Card, as indicated by Card AVN = '0002'), then instead of transitioning to **READING2** state, the Terminal returns to the **READY** state (Arrow 3), so it can process the 2<sup>nd</sup> tap of a 2-Tap transaction. The Terminal must re-use the same transaction context for the 2<sup>nd</sup> tap, and cache the authorization response so that it can be provided to the Card during the 2<sup>nd</sup> tap.

If the Terminal is processing the 2<sup>nd</sup> tap of a 2-Tap transaction, then the Terminal must ensure that the Card detected is the same Card used in the 1st tap. If the Card is a different, then the Terminal must request an AAC for the transaction to end the transaction. If the Card presented is the same card used in the 1<sup>st</sup> tap, and the GPO response corresponds to the transaction context of the 1st tap, then the Terminal must transition to **READING2** state immediately after performing Offline Data Authentication (Arrow 5).

When a Non-Legacy mode 2<sup>nd</sup> tap of 2-Tap transaction to be performed, the kernel will perform Callback Event Function **OnWaitCard(Flag=0x01)** immediately after **EXEP\_ucOnlineProcess** Event, and application notify to cardholder to 'present card again' to do 2<sup>nd</sup> tap of 2-Tap transaction.

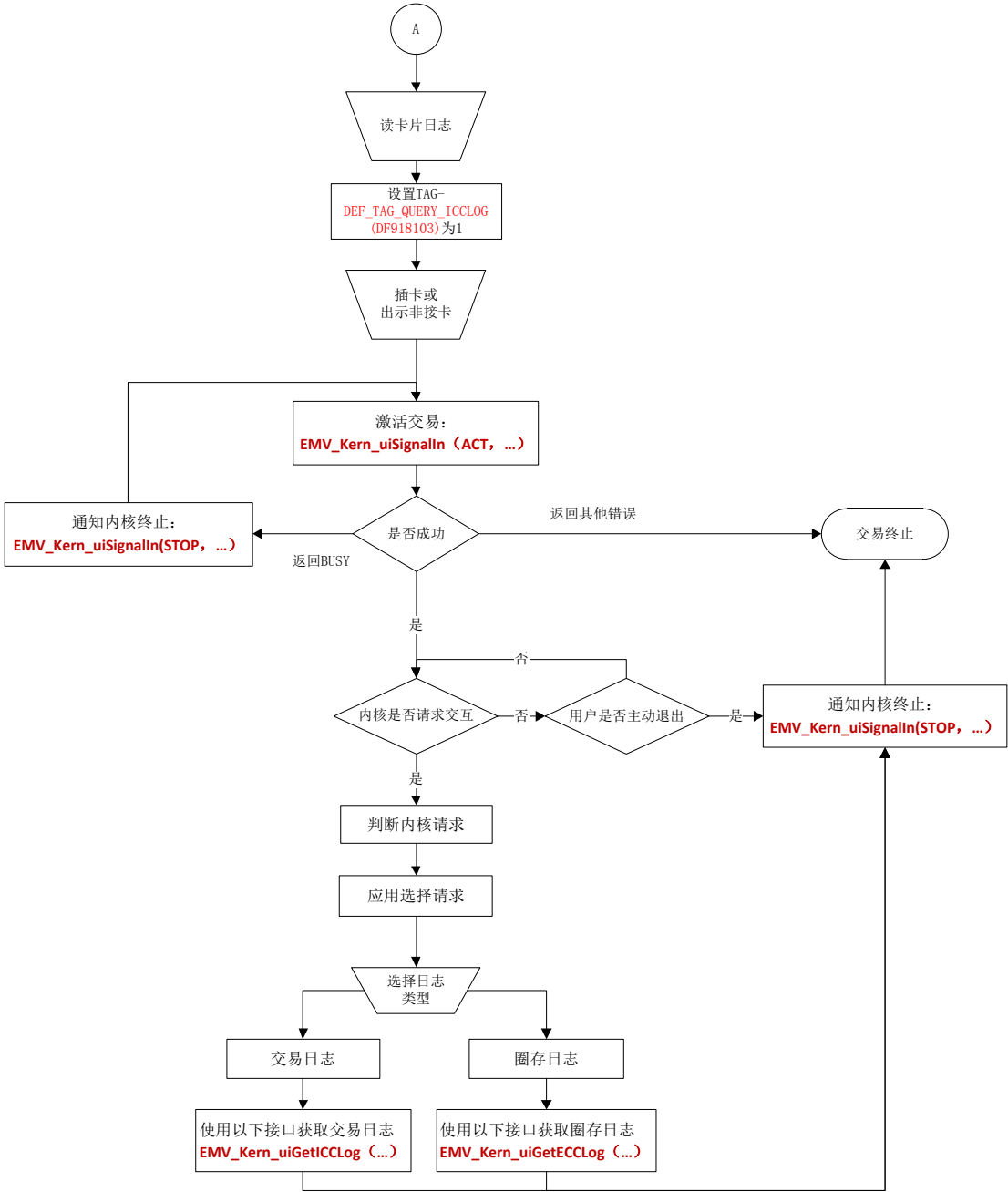
### 5.11.4 About Unblock Process(Cards with blocked application)

Both Non-Legacy and Legacy mode, the kernel will proceed GPO up to CAA1 when FINAL SELECTION the card response SW12=6283(means an application was blocked). Within an block application the card will response AAC(declined) to terminal while terminal required an ARQC/TC in CAA1 step. Then the kernel switch AAC to ARQC(online) perform Callback Event function **EXEP\_ucOnlineProcess** to go online within cryptogram data that

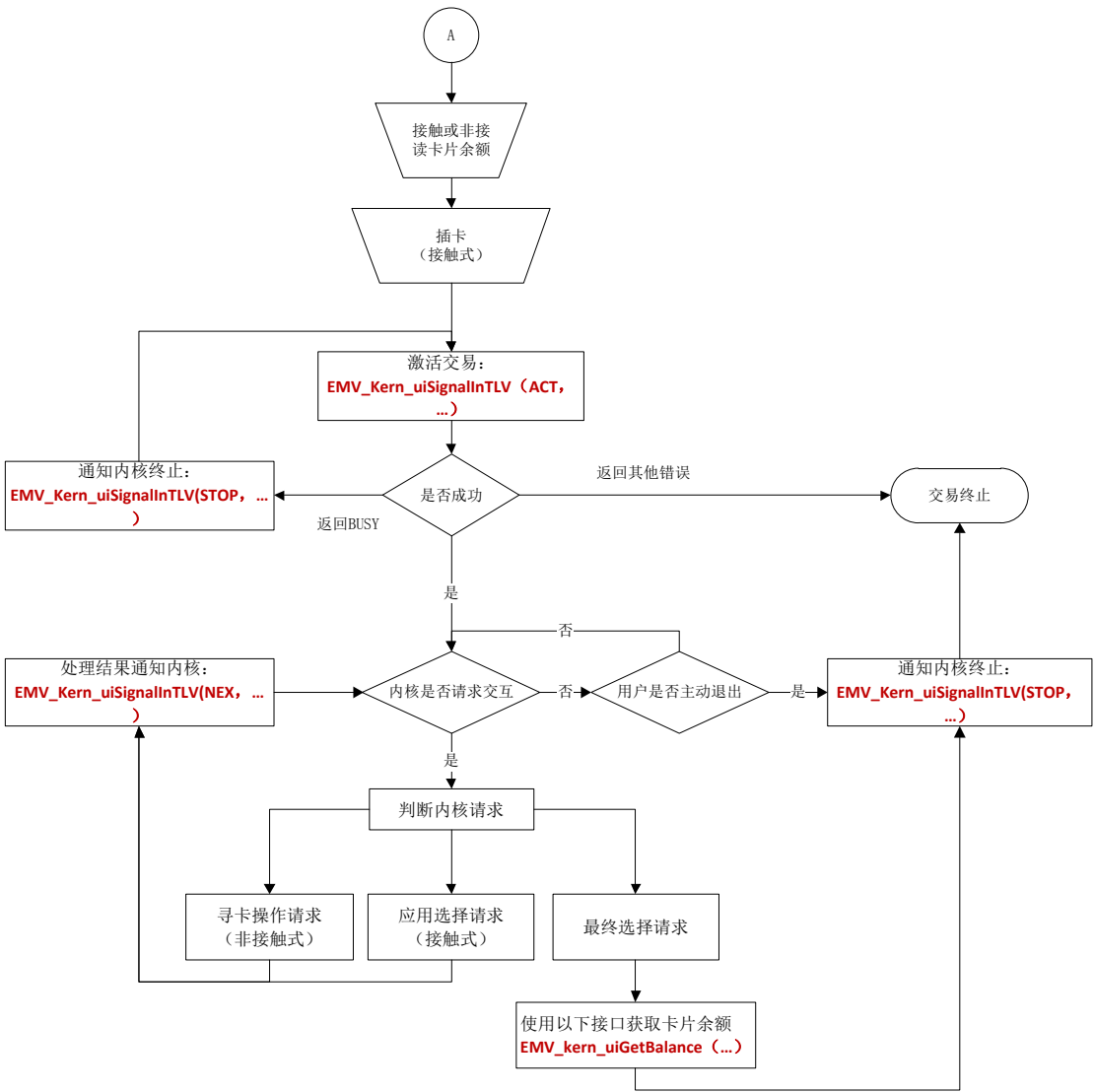


card generated response of CAA1 and terminal obtain issuer script from issuer response after online processing. Then the kernel will exchange APDU command within APPLICATION UNBLOCK to the card during Issuer Script Processing.

5.12 读卡片日志交易

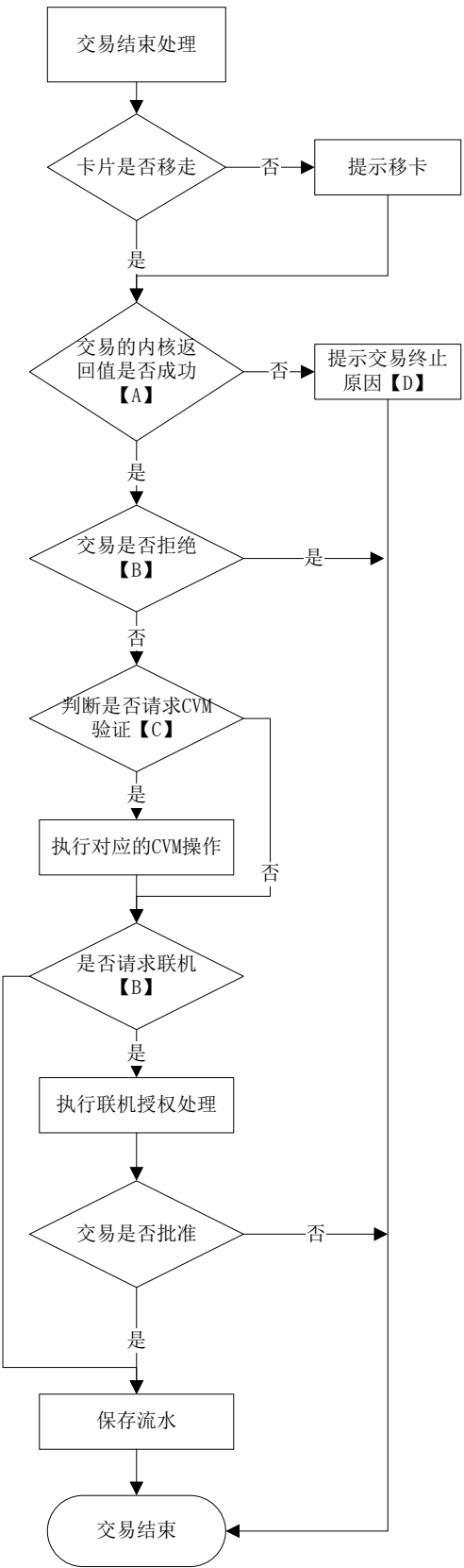


5.13 读卡片余额交易



### 5.14 交易结束处理

注释：  
【A】：判断结构EMV\_EXPAND\_INTERFACE中  
EXEP\_vEndProcess函数的第一个参数的值  
【B】：判断结构体EMV\_tTransData中的  
ucACType变量的值  
【C】：判断结构体EMV\_tTransData中的  
ucCVM变量的值  
【D】：具体的值对应的终止原因查看本文  
档中的第七章节的交易返回码



## 6. API 说明

### 6.1 EMV\_Kern\_uiCreateObject

API 原型	uint EMV_Kern_uiCreateObject(const EMV_Configuration *ptEMVConfiguration, EMV_OBJECT *ptEMVObject)		
API 功能	Create an EMV kernel Application Object.		
API 参数	类型	取值范围	参数说明
ptEMVConfiguration	入口参数	回调参数结构体指针	参考: <a href="#">EMV Configuration</a>
ptEMVObject	出口参数	将要创建对象的指针	
返回值	含义		
SUCCESS			创建成功
其他值			创建失败

**编程说明:**

该接口为 EMV 内核所有 API 中首个被调用的接口。回调函数接口同时也需要随同注册。内核会自动根据自身的需要动态分配内存空间。

6.2EMV\_Kern\_uiDestroyObject

API 原型	uint EMV_Kern_uiDestroyObject(EMV_OBJECT tEMVObject)		
API 功能	销毁已经创建成功的对象		
API 参数	类型	取值范围	参数说明
ptEMVObject	入口参数	将要销毁的对象的指针	该接口只是释放 tEMVObject 对象指向的内存空间，不会将其指向 NULL，应用调用该函数后需要将其指向 NULL，防止出现野指针！
返回值	含义		
SUCCESS	创建成功		
其他值	创建失败		

编程说明：

销毁的对象必须是之前已经创建成功的对象，如果未创建成功，仍然要销毁，将发生不可预知的结果。

### 6.3EMV\_Kern\_uiManageAID

<b>API 原型</b>	uint EMV_Kern_uiManageAID(EMV_OBJECT tEMVObject, uchar ucAction, uchar ucAIDLen, const uchar *pauAID, uchar ucPartSlt,uchar ucKID)		
<b>API 功能</b>	管理终端支持的 AID 列表		
<b>API 参数</b>	<b>类型</b>	<b>取值范围</b>	<b>参数说明</b>
tEMVObject	入口参数	应用的对象指针	
ucAction	入口参数	EMV_FLAG_ADD     -ADD AID EMV_FLAG_DELETE   -Delete AID EMV_FLAG_CLEAR     -Clear AID	动作标识 参考 <a href="#">附录 B 宏定义集合</a>
ucAIDLen	入口参数	5-16	AID 长度
pauAID	入口参数	Hex 数据串指针	AID 数据
ucPartSlt	入口参数	b8-b3: RFU b2:PPSE 失败转 AID 列表选择标识 1 - 支持 0 - 不支持 b1:部分选择标识 1 - 支持 0 - 不支持	b2:PPSE 失败转 AID 列表选择标识 b1:部分选择标识  默认不设置 b2 位，仅在 Discover ZIP 模式下设置 b2 位 1。
ucKID	入口参数	内核类型，支持的值有： 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-代表未指定 KernelID
<b>返回值</b>	<b>含义</b>		
SUCCESS	设置成功。		
0xF1	AID 长度错误。		
0xF2	AID 指针为空。		
0xF3	动作标识错误。		
0xF4	AID 存储空间已满（当前内核最多支持 100 个 AID）。		
0xF5	部分选择标识错误。		
其它值	其它错误。		

#### 编程说明：

启动 EMV 交易之前，应用必须设置终端所支持的 AID。当卡片 AID 与终端 AID 匹配后才能执行后续的 EMV 交易流程。

6.4EMV\_Kern\_uiUpdateCAIndexList

API 原型	uint EMV_Kern_uiUpdateCAIndexList(EMV_OBJECT tEMVObject, const uchar *pauRID, const uchar *pauIndexList, uchar ucListLen)		
API 功能	根据 RID，更新对应的 CA 公钥索引列表		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
pauRID	入口参数	Hex 数据串指针	CA 公钥 RID
pauIndexList	入口参数	Hex 数据串指针	CA 公钥索引列表
ucListLen	入口参数	0-50	CA 公钥索引列表长度 等于 0 时，清除该 RID 的公钥索引列表。
返回值	含义		
SUCCESS	设置成功。		
0x01	CA 公钥索引列表已满（当前内核最多可支持 6 组不同 RID 公钥索引列表）。		
0xF1	参数错误，RID 为空。		
0xF2	公钥索引列表长度超过最大值。		
其它值	其它错误。		

编程说明：

该接口针对非接触快速支付而设计，可以基本杜绝非接快速支付交易中因 CA 公钥缺失带来的卡片脱机余额扣款但交易失败的现象。对于支持 qPBOC 或者 qVSDC 的应用建议设置，如果应用未设置公钥索引列表，则内核将不执行预判处理，会提高因公钥缺失带来的交易失败且脱机扣款的风险。对于不支持非接触快速支付的应用可以不设置。



6.5EMV\_Kern\_uiSetCAPubKey

API 原型	uint EMV_Kern_uiSetCAPubKey(EMV_OBJECT tEMVObject, const EMV_tPKFILESTRU *ptPubKey)		
API 功能	设置 CA 公钥数据（RSA 算法公钥）		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ptPubKey	入口参数	结构 EMV_tPKFILESTRU 变量指针 或者 NULL	参考结构定义： <a href="#">EMV_tPKFILESTRU</a> 如果传入 NULL，将清空内核已存储的公钥数据。
返回值	含义		
SUCCESS	设置成功。		
0x01	公钥 Hash 值校验失败。		
0xF2	公钥数据格式错，请严格按照公钥结构定义的数据格式设置公钥数据。		
其它值	其它错误。		

编程说明：

该接口用于设置当前的 IC 卡交易所请求的 CA 公钥。应用在处理读卡片记录回调请求时，应根据内核反馈的公钥 RID 与公钥索引将对应的公钥数据传入给内核。

内核只能存储一组公钥数据，应用如果多次设置公钥数据，内核只保留最后一组。

6.6EMV\_Kern\_uiSetCAPubKey\_SM

API 原型	uint EMV_Kern_uiSetCAPubKey_SM(EMV_OBJECT tEMVObject, const EMV_tPKFILESTRU_SM *ptPubKey)		
API 功能	设置 CA 公钥数据（SM 算法公钥）		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ptPubKey	入口参数	结构 EMV_tPKFILESTRU_SM 变量 指针或者 NULL	参考结构定义： <a href="#">EMV tPKFILESTRU SM</a> 如果传入 NULL，将清空内核已存储的公钥数据。
返回值	含义		
SUCCESS	设置成功。		
0xF2	公钥数据格式错，请严格按照公钥结构定义的数据格式设置公钥数据。		
其它值	其它错误。		

编程说明：

该接口用于设置当前的 IC 卡交易所请求的 CA 公钥。应用在处理读卡片记录回调请求时，应根据内核反馈的公钥 RID 与公钥索引将对应的公钥数据传入给内核。

内核只能存储一组公钥数据，应用如果多次设置公钥数据，内核只保留最后一组。

6.7EMV\_Kern\_uiManageRecCert

API 原型	uint EMV_Kern_uiManageRecCert(EMV_OBJECT tEMVObject, uchar ucAction, const EMV_tRecCert *ptRecCert)		
API 功能	回收公钥证书管理		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ucAction	入口参数	EMV_FLAG_ADD -添加回 EMV_FLAG_DELETE -删除 EMV_FLAG_CLEAR -清空	动作标识，参考宏定义 Action Flag 当值为 EMV_FLAG_CLEAR 时， ptRecCert 可为空指针。
ptRecCert	入口参数	结构 EMV_tRecCert 变量指针	参考结构定义： <a href="#">EMV_tRecCert</a>
返回值	含义		
SUCCESS	操作成功。		
0x10	回收证书列表空间已满。		
0x11	添加的回收证书已经存在，重复添加。		
0xE1	参数错误，动作标识错误。		
0xE2	参数错误，回收证书指针为空。		
其它值	其它错误。		

编程说明：

该接口用于管理 EMV 内核维护的 CA 公钥回收证书列表，当前最多可存储 100 个。

6.8EMV\_Kern\_uiManageDOL

API 原型	uint EMV_Kern_uiManageDOL(EMV_OBJECT tEMVObject, uchar ucFlag, uchar ucActon, uchar *paDoL, uchar *pcLen)		
API 功能	终端缺省 DDOL、TDOL、UDOL 管理		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ucFlag	入口参数	1-DDOL 2-TDOL 3-UDOL	DOL 类型
ucAction	入口参数	0x01-设置 DOL 值 0x02-获取 DOL 值	动作标识。
paDoL	入口\出口 参数	DOL 值域变量指针	当设置 DOL 时, DOL 的值域(入口) 当获取 DOL 时, 存储 DOL 的值域(出口)
pcLen	入口\出口 参数	DOL 值域长度变量指针 当设置 DOL 时, 取值范围 0-100	当设置 DOL 时, DOL 的值域的长度(入口) 当获取 DOL 时, 存储 DOL 值域的长度(出口)
返回值	含义		
SUCCESS	操作成功。		
0x01	参数错误, DOL 值域及其长度指针为空。		
0x02	参数错误, 动作标识错误。		
0x03	参数错误, DOL 类型错误。		
0x11/0x21/0x31	设置 DOL 长度超过最大长度。		
0x12/0x22/0x32	DOL 数据格式错误。		
其它值	其它错误。		

编程说明:

该接口用于设置终端缺省 DDOL、TDOL、UDOL。

例如:

设置 DDOL="9F3704":

```
Uchar ucLen;  
ucLen=3;  
uiRet = EMV_Kern_uiManageDOL(1, 1, (uchar*)"9F3704", &ucLen);
```

获取 DDOL:

```
Uchar ucLen;  
Uchar auBuffer;  
ucLen=3;  
uiRet = EMV_Kern_uiManageDOL(1, 2, auBuffer, &ucLen);
```

### 6.9 EMV\_Kern\_uiSignalInTLV

API 原型	uint EMV_Kern_uiSignalInTLV(EMV_OBJECT tEMVObject, uchar ucSignal, const uchar *pauTLVData, uint uiTLVDataLen)		
API 功能	EMV 交易信号输入		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ucSignal	入口参数	EMV_SIGNAL_ACT EMV_SIGNAL_NEXT EMV_SIGNAL_STOP EMV_SIGNAL_CLEAN	信号类型 参考 <a href="#">附录 B 宏定义集合</a>
pauTLVData	入口参数	NULL 或者 TLV 参数数据指针	TLV 参数数据 不同阶段需传入不同的参数，参考 <a href="#">表格 4-1</a> 。
uiTLVDataLen	入口参数	0 或者 TLV 参数数据长度	TLV 参数数据长度
返回值	含义		
SUCCESS	处理成功。		
EMV_RESULT_BUSY	EMV 工作线程忙碌中，请稍后再尝试。		
0xF001	信号标识错误，交易未激活不能发送 NEXT 信号。		
0xF0F1	参数数据不符合 TLV 格式。		
0xF1XX	信号标识错误，未定义的信号标识 XX。		
0xF201	信号参数长度错误。		
其它值	其它错误。		

#### 编程说明：

例如：开启一笔新的 EMV 交易

```
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>API 原型</b>	uint EMV_Kern_uiSetTLV(EMV_OBJECT tEMVObject, uchar ucKernelID, const uchar *paTag, uchar ucValueLen, const uchar *paValue)		
<b>API 功能</b>	设置单个 TLV 数据元		
<b>API 参数</b>	<b>类型</b>	<b>取值范围</b>	<b>参数说明</b>
tEMVObject	入口参数	应用的对象指针	
ucKernelID	入口参数	参考 <a href="#">内核类型宏定义</a>	可根据标签宏定义的前缀来辨别内核类型： 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) 参考 <a href="#">附录 A 标签定义</a>
paTag	入口参数	Hex 数据串指针	要设置的标签
ucValueLen	入口参数	0 或者 1~0xFF	要设置的值的长度, 0 代表清空数据元
paValue	入口参数	NULL, 当且仅当 ValueLen 等于 0 或者 Hex 数据串指针	要设置的值, NULL 代表清空数据元
<b>返回值</b>	<b>含义</b>		
SUCCESS	设置成功。		
0x00F0	参数错误		
0x00F1	IC 卡数据元标签, 不可设置		
0x00F2	不可识别内核类型		
0x00E1	数据格式错误		
其它值	其它错误		

### 编程说明:

例如:

设置 POS 输入点方式:

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

设置 Mobile Support Indicator

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

清空终端序列号:

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

### 6.11 EMV\_Kern\_uiSetTLVList

API 原型	uint EMV_Kern_uiSetTLVList(EMV_OBJECT tEMVObject, uchar ucKernelID, const uchar *pauData, uint uiDataLen)		
API 功能	设置 TLV 数据元列表		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ucKernelID	入口参数	参考内核类型宏定义	<p>可根据标签宏定义的前缀来辨别内核类型：</p> <p>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)</p> <p>参考<a href="#">附录 A 标签定义</a></p> <p><b>说明：</b></p> <p>内核类型为 PBOC、VISA、MasterCard 定义的标签不能在同一个 TLV 数据串中设置，在此前提，做如下判断：</p> <ul style="list-style-type: none"><li>• 如果 TLV 数据串中包含 PBOC 标签，则内核类型应设置为 EMV_KERNELID_PBOC</li><li>• 如果 TLV 数据串中包含 VISA 标签，则内核类型应设置为 EMV_KERNELID_VISA</li><li>• 如果 TLV 数据串中包含 MASTER 标签，则内核类型应设置为 EMV_KERNELID_MASTER</li><li>• 如果 TLV 数据串中包含 AMEX 标签，则内核类型应设置为 EMV_KERNELID_AMEX</li><li>• 如果 TLV 数据串中包含 DISCOVER 标签，则内核类型应设置为 EMV_KERNELID_DISCOVER</li><li>• 如果 TLV 数据串中包含 JCB 标签，则内核类型应设置为 EMV_KERNELID_JCB</li><li>• 如果 TLV 数据串中仅包含 EMV 或（与）DEFINE 标签，则内核</li></ul>

			类型设置为： EMV_KERNELID_EMV 或者 EMV_KERNELID_DEFINE
pauData	入口参数	Hex 数据串指针	TLV 格式数据串
uiDataLen	入口参数		数据串的长度
返回值	含义		
SUCCESS	设置成功。		
其它值	其它错误。		

**编程说明：**

例如：

**情况一：**只包含 EMV 定义标签 **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);



情况二：只包含 DEFINE 定义标签 **DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

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

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

情况三：同时包含 EMV 和 DEFINE 定义标签 **EMV\_TAG\_TM\_CNTRYCODE, EMV\_TAG\_TM\_CURCODE,**

**DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

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

情况四：同时包含 PBOC 与 EMV 与(或)DEFINE 定义标签 **C\_TAG\_TM\_9F7A, EMV\_TAG\_TM\_CURCODE,**

**DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

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

情况五：同时包含 VISA 与 EMV 与(或)DEFINE 定义标签 **V\_TAG\_TM\_9F66, EMV\_TAG\_TM\_CURCODE,**

**DEF\_TAG\_PSE\_FLAG, DEF\_TAG\_QUERY\_ICCLOG**

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

情况六：同时包含 MASTER 与 EMV 与(或)DEFINE 定义标签 **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

API 原型	uint EMV_Kern_uiGetTLV(EMV_OBJECT tEMVObject, const uchar *pTag, uchar ucTagLen, uchar *pVal, uint *puiLen)		
API 功能	获取单个 Tag 的值		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
pTag	入口参数	Hex 数据串指针	要获取的数据的标签
ucTagLen	入口参数	标签长度	要获取的数据的标签的长度
pVal	出口参数	Hex 数据串指针	存放获取数据的缓存
puiLen	出口参数	值域长度指针	获取到的数据长度
返回值	含义		
SUCCESS	获取成功。		
0x01	该标签未被赋值。		
0xE1	不可识别标签。		
0xE2	标签错误。		
其它值	其它错误。		

**编程说明:**

例如，完成一笔交易后，从内核获取主账号：

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

6.13 EMV\_kern\_uiGetBalance

API 原型	uint EMV_kern_uiGetBalance(EMV_OBJECT tEMVObject, uchar *pauBalance, uchar *pucLen)		
API 功能	通过向卡片发送 APDU 指令的方式获取接触或非接卡片可用余额		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
pauBalance	出口参数		存放获取余额的缓存
pucLen	出口参数		余额数据的长度
返回值	含义		
SUCCESS	获取成功。		
其它值	其它错误或者失败。		

编程说明:

该接口调用时机，需要在最终选择事件产生后。

6.14 EMV\_kern\_uiGetDataAPDU

API 原型	uint EMV_kern_uiGetDataAPDU(EMV_OBJECT tEMVObject, uchar ucP1, uchar ucP2, uchar *pauValue, ushort *pusValueLen)		
API 功能	通过发送 APDU 指令获取卡片内部数据元		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ucP1	入口参数		标签首字节
ucP2	入口参数		标签次字节
pauValue	出口参数		获取数据元的值域
pusValueLen	出口参数		获取数据元的值域的长度
返回值	含义		
SUCCESS			获取成功。
其它值			其它错误。

编程说明:

该接口调用时机，需要在最终选择事件产生后。

例如获取卡片交易计数器（ATC）：

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

6.15 EMV\_Kern\_uiGetICCLog

API 原型	uint EMV_Kern_uiGetICCLog(EMV_OBJECT tEMVObject, const EMV_tSelectAID *ptSelectedAID, uchar *pucLogNum, EMV_tICCLog *ptICCLog)		
API 功能	获取 IC 卡卡片交易日志		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ptSelectedAID	入口参数	结构 EMV_tSelectAID 变量指针	参考结构定义: <a href="#">EMV_tSelectAID</a>
pucLogNum	入口参数	0-0x50	应用期望获取日志的最大数量
pucLogNum	出口参数	0-0x50	实际获取到的日志的数量
ptICCLog	出口参数	结构 EMV_tICCLog 变量指针	参考结构定义: <a href="#">EMV_tICCLog</a>
返回值	含义		
SUCCESS	获取成功。		
0x01	卡片不支持日志获取。		
其它值	其它错误。		

编程说明:

该接口调用时机, 需要在应用选择事件产生后。

6.16EMV\_Kern\_uiGetECCLog

API 原型	uint EMV_Kern_uiGetECCLog(EMV_OBJECT tEMVObject, const EMV_tSelectAID *ptSelectedAID, uchar *pucLogNum, EMV_tECCLog *ptECCLog)		
API 功能	获取 IC 卡卡片圈存日志		
API 参数	类型	取值范围	参数说明
tEMVObject	入口参数	应用的对象指针	
ptSelectedAID	入口参数	结构 EMV_tSelectAID 变量指针	参考结构定义: <a href="#">EMV_tSelectAID</a>
pucLogNum	入口参数	0-0x50	应用期望获取日志的最大数量
pucLogNum	出口参数	0-0x50	实际获取到的日志的数量
ptECCLog	出口参数	结构 EMV_tECCLog 变量指针	参考结构定义: <a href="#">EMV_tECCLog</a>
返回值	含义		
SUCCESS	获取成功。		
0x01	卡片不支持日志获取。		
其它值	其它错误。		

**编程说明:**  
该接口调用时机，需要在应用选择事件产生后，并且针对支持圈存交易的卡片。例如 PBOC3.0 电子现金卡。

## 6.17 EMV\_Kern\_vSwitchDebug

<b>API 原型</b>	void EMV_Kern_vSwitchDebug(EMV_OBJECT tEMVObject, uchar ucMode);		
<b>API 功能</b>	Deactive or active kernel debug log function.		
<b>API 参数</b>	<b>类型</b>	<b>取值范围</b>	<b>参数说明</b>
tEMVObject	入口参数	应用的对象指针	
ucMode	入口参数	0 –关闭 Debug 功能 1 –打开 Debug 功能，并且在交易结束后获取 Debug 日志 2 –打开 Debug 功能，并且在交易过程中实时输出日志	参考: <a href="#">获取内核 Debug 日志</a>
<b>返回值</b>	<b>含义</b>		
void	SUCCESS.		

编程说明:

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

**6.18 EMV\_kern\_uiSetHandle**

<b>API 原型</b>	uint EMV_kern_uiSetHandle(EMV_OBJECT tEMVObject, uchar ucFlag, EMVHandle hHandle)		
<b>API 功能</b>	设置 IC 卡读卡设备或者 pinpad 设备句柄		
<b>API 参数</b>	<b>类型</b>	<b>取值范围</b>	<b>参数说明</b>
tEMVObject	入口参数	应用的对象指针	
ucFlag	入口参数	0xA0 –接触式 IC 卡读卡器 0xA1 –内置非接 IC 卡读卡器 0xA2 –外置非接 IC 卡读卡器 0xB1 – PinPad	
hHandle	入口参数	EMVHandle	设备句柄
<b>返回值</b>	<b>含义</b>		
uint	SUCCESS.		
其它值	未定义标识.		

**编程说明:**



# 7. 结构定义

## EMV\_Configuration

结构名称	注册回调函数参数结构	
参数类型	参数名称	参数说明
EMV_EXPAND_BASEFUN	tExAPI_BASE	参考: <a href="#">EMV_EXPAND_BASEFUN</a>
EMV_EXPAND_INTERFACE	tExAPI_IFC	参考: <a href="#">EMV_EXPAND_INTERFACE</a>
uint	uiConfigDataLen	配置参数长度
uchar	auConfigData[600]	配置参数
uchar	ucConfigID	EMV 配置 ID
uchar	ucDebugMode	0-关闭日志，1-交易完成后获取日志，2-交易过程中实时输出日志

EMV\_tPKFILESTRU

结构名称	RSA 算法 CA 公钥结构	
参数类型	参数名称	参数说明
uchar	auRid[5];	RID
uchar	ucIndex;	公钥索引
uchar	ucModLen;	公钥模的长度
uchar	auMod[256];	公钥模
uchar	ucExpLen;	公钥指数的长度：1 或者 3
uchar	auExp[3];	公钥指数：\x03 或者 \x01\x00\x01
uchar	auExpDate[4];	公钥失效期, YYYYMMDD
uchar	ucHashFlg;	公钥校验和是否存在：1-存在 0-不存在，当为 0 时，不进行哈希校验
uchar	auHash[20];	公钥校验和 - 用于校验公钥数据是否正确，可不赋值

**EMV\_tPKFILESTRU\_SM**

结构名称	SM 算法 CA 公钥结构	
参数类型	参数名称	参数说明
uchar	auRid[5];	RID
uchar	ucIndex;	公钥索引
uchar	ucKeyLen;	公钥长度
uchar	auPubKey[256];	公钥数据
uchar	auExpDate[4];	公钥失效期, YYYYMMDD
GROUP_PARA	tGroupPara	椭圆曲线参数（缺省填 NULL）

EMV\_tSelectAID

结构名称	持卡人应用选择所选中的 AID	
参数类型	参数名称	参数说明
uchar	ucAIDLen;	被选中的 AID 长度
uchar	auAID[EMV_LEN_MAX_AID];	被选中的 AID 数据

EMV\_tRecCert

结构名称	回收公钥证书	
参数类型	参数名称	参数说明
uchar	ucIndex;	公钥索引
uchar	auRID[5];	RID
uchar	auSN[3];	证书序列号

## EMV\_tICCLog

结构名称	IC 卡交易日志	
参数类型	参数名称	参数说明
uchar	ucAmtFlg;	交易金额存在标识: 0-不存在 1-存在
uchar	auAmount[6];	交易金额: BCD 编码
uchar	ucAmtOthFlg;	其它金额存在标识: 0-不存在 1-存在
uchar	auAmountOth[6];	其它金额: BCD 编码
uchar	ucDateFlg;	交易日期存在标识: 0-不存在 1-存在
uchar	auDate[3];	交易日期: BCD 编码
uchar	ucTimeFlg;	交易时间存在标识: 0-不存在 1-存在
uchar	auTime[3];	交易时间: BCD 编码
uchar	ucCntCFlg;	国家代码存在标识: 0-不存在 1-存在
uchar	auCntCode[2];	国家代码(9F1A)
uchar	ucCurCFlg;	货币代码存在标识: 0-不存在 1-存在
uchar	auCurCode[2];	货币代码(5F2A)
uchar	ucATCFlg;	交易计数器存在标识: 0-不存在 1-存在
uchar	auATC[2];	交易计数器(9F36)
uchar	ucSevFlg;	交易类型存在标识: 0-不存在 1-存在
uchar	ucServeType;	交易类型(9C)
uchar	ucMchFlg;	商户名称存在标识: 0-不存在 1-存在
char	szMchName[30];	商户名称(9F4E)
uchar	ucTLVLen;	TLV 数据长度
uchar	auTLV[256];	本结构中未定义的有关数据元按照 TLV 列表的格式保存在 auTLV 中

## EMV\_tECCLog

结构名称	IC 卡圈存日志	
参数类型	参数名称	参数说明
uchar	auECTag[2];	卡片余额对应的标签
uchar	auPreValue[6];	修改前余额
uchar	auAftValue[6];	修改后余额
uchar	ucDateFlg;	交易日期存在标识：0-不存在 1-存在
uchar	auDate[3];	交易日期：BCD 编码
uchar	ucTimeFlg;	交易时间存在标识：0-不存在 1-存在
uchar	auTime[3];	交易时间：BCD 编码
uchar	ucCntCFlg;	国家代码存在标识：0-不存在 1-存在
uchar	auCntCode[2];	国家代码(9F1A)
uchar	ucATCFlg;	交易计数器存在标识：0-不存在 1-存在
uchar	auATC[2];	交易计数器(9F36)
uchar	ucMchFlg;	商户名称存在标识：0-不存在 1-存在
char	szMchName[30];	商户名称(9F4E)
uchar	ucTLVLen;	TLV 数据长度
uchar	auTLV[50];	本结构中未定义的其它数据元按照 TLV 列表的格式保存在 auTLV 中

## EMV\_tCandAIDInfo

结构名称	候选 AID 信息	
参数类型	参数名称	参数说明
uchar	ucAIDLen;	
uchar	auAID[EMV_LEN_MAX_AID];	AID
uchar	ucLabelLen;	应用标签长度
uchar	auAppLabel[16];	应用标签
uchar	ucAPNLen;	应用首选名长度
uchar	auAPN[16];	应用首选名
uchar	ucAPIIDFlag;	应用优先权标识符是否存在：0-不存在 1-存在
uchar	ucAPIID;	应用优先权标识符
uchar	ucLangPrefLen;	首选语言长度
uchar	auLangPref[8];	首选语言
uchar	ucIssCTIndexFlag;	发卡行代码表索引是否存在：0-不存在 1-存在
uchar	ucIssCTIndex;	发卡行代码表索引
uchar	ucKernelIDLen;	Kernel ID 长度
uchar	auKernelID[8];	kernel ID



EMV\_tAIDCandList

结构名称	应用选择候选 AID 信息列表	
参数类型	参数名称	参数说明
uchar	ucReSelectFlag;	是否重新应用选择:0-否 1-是
uchar	ucCandAIDNum;	候选 AID 个数
EMV_tCandAIDInfo	patCandList	候选 AID 信息列表

EMV\_tFinalData

结构名称	最终选择数据	
参数类型	参数名称	参数说明
uchar	ucAIDLen;	被选中的 AID 长度
uchar	auAID[EMV_LEN_MAX_AID];	被选中的 AID 数据
uchar	ucKernelID;	Kernel ID
uchar	ucPIDLen;	VISA:Program ID 长度
uchar	auPID[16];	VISA:Program ID

**EMV\_tRecordData**

结构名称	读卡片记录返回数据	
参数类型	参数名称	参数说明
uchar	ucAIDLen;	
uchar	auAID[EMV_LEN_MAX_AID];	AID
uchar	ucPanLen;	主账号长度
uchar	auPan[10];	主账号（BCD 码）
uchar	ucPanSNFlag;	主账号序列号标识: 0-不存在
uchar	ucPanSN;	主账号序列号
uchar	auExpiry[4];	应用失效日期（BCD 码）
uchar	ucAlgorithmID;	算法标识：00-RSA    01-SM
uchar	ucPubKIndex;	公钥索引
uchar	ucFlowType;	交易流程标识:参考交易流程标识宏定义
uchar	auECIAC[6];	电子现金发卡行授权码(9F74)
uchar	ucSFI11;	对于 paywave2 交易，判断是否包含 SFI11
uchar	RFULen;	
uchar	RFU[256];	

EMV\_tCVM

结构名称	持卡人验证方法	
参数类型	参数名称	参数说明
uchar	ucCVM;	CVM 标识: 参考宏定义
uchar	ucPINTimes;	脱机 PIN 剩余尝试次数: 0-卡片未返回重试次数
uchar	ucCertType;	证件类型(证件类型: 00-身份证 01-军官证 02-护照 03-入境证 04-临时身份证 05-其他)
uchar	ucCertLen;	证件号码长度
uchar	auCert[40];	证件号码

**EMV\_tDisplayMsg**

结构名称	交互信息显示结构	
参数类型	参数名称	参数说明
uchar	ucMsgID;	Message ID ： 参考宏定义
uchar	ucCurrency ;	货币标识 0-人民币 1-美元
uchar	ucDataLen;	信息长度
uchar	auData[30];	信息数据

EMV\_tErrorID

结构名称	交易开始参数	
参数类型	参数名称	参数说明
uchar	ucL1;	L1 层错误，参考宏定义:错误类型定位标识
uchar	ucL2;	L2 层错误，参考宏定义:错误类型定位标识
uchar	ucL3;	L3 层错误，参考宏定义:错误类型定位标识
uchar	ucMsgID;	Message ID，参考宏定义:信息标识 Message ID
ushort	usSW12;	APDU 卡片返回状态码

EMV\_tTransData

结构名称	交易结束数据	
参数类型	参数名称	参数说明
uchar	ucACType;	密文类型:0-拒绝 1-批准 2-联机请求
uchar	ucCVM;	CVM 标识: 参考宏定义
uchar	ucPanLen;	主账号长度
uchar	auPan[10];	主账号
uchar	ucPanSnFlag;	是否存在主账户序列号
uchar	ucPanSn;	持卡人主账户序列号。
uchar	auExpiry[3];	应用失效日期
uchar	ucFlowType;	交易流程标识:参考交易流程标识宏定义
uchar	ucECIACFlag;	是否存在电子现金发卡行授权码
uchar	auECIAC[6];	电子现金发卡行授权码(9F74)
uchar	ucBalanceFlag;	是否存在余额
uchar	auBalance[6];	如果余额存在, 保存余额数据 (BCD 编码)
uchar	ucCID;	密文信息数据
uchar	ucSRLength;	脚本执行结果长度
uchar	auScriptResult[100];	脚本执行结果
uchar	ucMSDT1Len;	MSD 交易一磁道模拟数据长度
uchar	auMSDT1Data[200];	MSD 交易一磁道模拟数据
uchar	ucMSDT2Len;	MSD 交易二磁道模拟数据长度
uchar	auMSDT2Data[100];	MSD 交易二磁道模拟数据
uint	uiTLVLen;	TLV 数据元长度
uchar	auTLVData[300];	TLV 数据元

EMV\_EXPAND\_BASEFUN

结构名称	驱动层回调函数接口列表	
参数类型	参数名称	参数说明
函数指针	EX_API_ucICCIO	执行 APDU 交互
函数指针	EX_API_ucListener	内核监听
函数指针	EX_API_ucGetRandomData	产生随机数
函数指针	EX_API_ucRSACal	RSA 计算
函数指针	EX_API_ucCalculateHash	Hash/SHA1 算法计算
函数指针	EX_API_ucCalculateHash_SM	国密 Hash 值计算
函数指针	EX_API_ucVerifySign_SM	国密签名验证
函数指针	EX_API_ucVerifyOfflinePIN	脱机 PIN 验证

回调函数指针原型：

```
uchar (*EX_API_ucICCIO)(EMV_tICCCDev tICCCDevice, uint uiSendLen, const void* pvDataIn, uint *puiRecLen, void* pvDataOut);
uchar (*EX_API_ucListener)(uchar ucFlag, EMVHandle hMagDevice); //Flag:1-磁条卡 2-接触 IC 卡
uchar (*EX_API_ucGetRandomData)(uint uiLen, uchar *pauData);
uchar (*EX_API_ucRSACal)(uint uiInLen, 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_SM *ptPubKey, uchar *pauHash);
uchar (*EX_API_ucVerifySign_SM)(const uchar *pauHash, uint uiLen, const uchar *pauInData, const EMV_tPKFILESTRU_SM *ptPubKey);
uchar (*EX_API_ucVerifyOfflinePIN)(uchar ucFlag, const uchar *pauRandom, const EMV_tPKFILESTRU *ptPubKey, ushort *pauSW12); //Flag:0-脱机明文 PIN 1-脱机密文 PIN
```

参考: [EMV Configuration](#).



## EMV\_EXPAND\_INTERFACE

结构名称	交互界面 API 回调列表	
参数类型	参数名称	参数说明
函数指针	EXEP_ucWaitCard	请求非接触寻卡处理
函数指针	EXEP_ucAppSelection	请求持卡人应用选择处理
函数指针	EXEP_ucFinalSlt	最终选择后处理，本阶段可根据 AID 设置交易参数
函数指针	EXEP_ucReadRecord	读卡片记录完成后处理，本阶段可显示卡号，查询黑名单等
函数指针	EXEP_ucCardHolderVerify	请求持卡人验证处理
函数指针	EXEP_ucOnlineProcess	请求联机交易处理
函数指针	EXEP_vEndProcess	通知交易结束处理
函数指针	EXEP_vSendOut	内核输出 (无需上层回应)
函数指针	EXEP_vObtain	内核输入 (等待上层回应，并发 NEXT 信号继续)

### 回调函数指针原型：

```
//非接寻卡处理
uchar (*EXEP_ucWaitCard)(uchar ucFlag);

//应用选择返回处理
uchar (*EXEP_ucAppSelection)(const EMV_tAIDCandList *ptDCandList, EMV_tSelectAID *ptSelectedAID4);

//最终选择返回处理
uchar (*EXEP_ucFinalSlt)(const EMV_tFinalData *ptFinalData, EMV_tGPOParam *ptDataBack5);

//读记录完成后处理
uchar (*EXEP_ucReadRecord)(const EMV_tRecordData *ptRecordData, EMV_tTRManage *ptTRManage6);

//持卡人验证
uchar (*EXEP_ucCardHolderVerify)(const EMV_tCVM *ptCVM, EMV_tCHVData *ptCHVData7);

//联机处理
uchar (*EXEP_ucOnlineProcess)(const EMV_tTransData *ptTransData, EMV_tHostData *ptHostData8);

//交易结束处理
void (*EXEP_vEndProcess)(uint uiResult, const EMV_tTransData *ptTransData);

//内核输出 (无需上层回应)
void (*EXEP_vSendOut)(uchar ucINS, uint uiDataLen, const uchar *pauData);

//内核输入 (等待上层回应，并发 NEXT 信号继续)
void (*EXEP_vObtain)(uchar ucINS, uint uiDataLen, const uchar *pauData);
```

### 说明：

以上回调函数指针原型中，应用具体在实现这些回调函数的时候，**绿色**部分的参数为入口参数，也就是说，回调函数实现时，应用端需存储的参数，回调函数的实现仅限于存储入口参数，保存后应该立即退出，请参照 API 说明 [EMV Configuration](#) 的编程指南。

**黄色**部分参数为出口参数，但要注意的是，在回调函数实现时，不需要对该参数做任何处理，这些参数是在 [EMV Kern.uiSignalInTLV](#) (NEXT, ...) 时，作为入口参数传递给内核。黄色部分参数设计在

<sup>4</sup>多应用接口已经删除该参数。

<sup>5</sup>多应用接口已经删除该参数。

<sup>6</sup>多应用接口已经删除该参数。

<sup>7</sup>多应用接口已经删除该参数。

<sup>8</sup>多应用接口已经删除该参数。

回调参数中没有实际意义，只起到参考作用，方便程序员了解在某个回调请求后，需要在 EMV\_Kern\_uiSignalInTLV（NEXT，...）时，传递给内核什么样的参数。

### 编程说明：

EMV 交易回调 API 列表的注册，是启动 EMV 交易前的必备操作。回调 API 列表结构中的**所有回调 API 的内部实现不允许有阻塞，耗时的操作，应遵循操作越简单快速越好的原则**，否则 EMV 工作线程将因此被阻塞在回调函数中，降低了异步操作的时效性能，增加了 **EMV\_Kern\_uiSignalInTLV** 返回 EMV\_RESULT\_BUSY 的几率。

**例如：**EMV 内核回调请求 R1， R1 的回调函数内部实现如下：

- 1、 保存内核反馈给应用的参数
- 2、 标记内核请求的交互类型 **R1**
- 3、 结束处理，返回成功

**读记录回调实现的示例代码：**

```
uchar EG_ucSignal;           //交互类型标识
uchar EG_auSignalData[1000]; //回调入口参数
uint  EG_uiSignalDataLen;    //回调入口参数长度

//读记录回调 API 的实现如下：
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;
    return EM_SUCCESS;
}
```

## 8. 获取内核 Debug 日志

在开发或者维护过程中，如果发现内核处理存在问题或者异常，应用应该尝试获取内核的交易日志，提供给 EMV 维护人员分析。具体获取方法如下步骤所示：

1. 在 EMV 内核初始化后，并且在开始一笔 EMV 交易之前，应用通过调用 `EMV_Kern_vSwitchDebug` 来设置获取 Debug 日志的模式。目前有 3 种 debug 模式，分别为：0-关闭日志，1-交易完成后获取日志，2-交易过程中实时输出日志。
2. 如果设置模式 0-关闭日志，则内核不输出任何 Debug 日志信息。
3. 如果设置模式 1-交易完成后获取日志，内核在交易结束后，会将日志信息通过 `EMV_EXPAND_INTERFACE.EXEP_vSendOut` 回调函数传递给应用。应用需判断回调函数的第一个参数 `ucINS` 的值，如果值为 `EMV_INS_DBLOG`，则第三个参数就是内核传出的日志。EMV 日志信息都是 ASCII 可视字符，应用可通过串口或者其它手段将日志数据转发到 PC，并且在 PC 端通过串口工具接收 ASCII 字符信息，即可完成 EMV 日志的获取。
4. 如果设置模式 2-交易过程中实时输出日志，内核会在交易过程中，实时的通过 `EMV_EXPAND_INTERFACE.EXEP_vSendOut` 回调函数传递给应用。应用需判断回调函数的第一个参数 `ucINS` 的值，如果值为 `EMV_INS_DBLOG`，则第三个参数就是内核传出的日志。EMV 日志信息都是 ASCII 可视字符，应用可通过串口或者其它手段将日志数据转发到 PC，并且在 PC 端通过串口工具接收 ASCII 字符信息，即可完成 EMV 日志的获取。

## 附录 A 标签定义列表

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

Kernel ID	EMV_KERNELID_PBOC				
Tag Name	Definition	Description			
EMV_TAG_TM_TERMTYPE	Tag: 9F35 Length: 1 Format:BCD Kernel: EMV	Terminal Type			
		Indicates the environment of the terminal, itscommunications capability, and its operationalcontrol.			
		Environment	Operator		
			Financial Institution	Mechant	Cardholder
		Attendant:			
		Online only	11	21	-
		Online and Offline	12	22	-
		Offline only	13	23	-
		Self-help:			
		Online only	14	24	34
Online and Offline	15	25	35		
Offline only	16	26	36		
EMV_TAG_TM_CAP	Tag: 9F33 Length: 3 Format:b Kernel: EMV	Terminal Capabilities Indicates the card data input, CVM, andsecurity capabilities of the terminal			
EMV_TAG_TM_CAP_AD	Tag: 9F40 Length: 5 Format:b Kernel: EMV	Additional Terminal Capabilities Indicates the data input and outputcapabilities of the terminal			
EMV_TAG_TM_CNTRYCODE	Tag: 9F1A Length: 2 Format:b Kernel: EMV	Terminal Country Code Indicates the country of the terminal,represented according to ISO 3166			
EMV_TAG_TM_CURCODE	Tag: 5F2A Length: 2 Format:b Kernel: EMV	TransactionCurrency Code Indicates the currency code of the transactionaccording to ISO 4217			
EMV_TAG_TM_FLOORLMT	Tag: 9F1B Length: 4 Format:b Kernel: EMV	Terminal Floor Limit Indicates the floor limit in the terminal inconjunction with the AID			
EMV_TAG_TM_AID	Tag: 9F06 Length: 5-16 Format: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: 9F7A</b> <b>Length: 1</b> <b>Format:b</b> <b>Kernel: PBOC</b>	Indicate supporting Ecash or not. 0 - No 1 - Yes																																		
C_TAG_TM_DF69	<b>Tag: DF69</b> <b>Length: 1</b> <b>Format:b</b> <b>Kernel: PBOC</b>	Indicate supporting PBOC SM algorithm or not. 0 - No 1 - Yes																																		
C_TAG_TM_9F66	<b>Tag: 9F66</b> <b>Length: 4</b> <b>Format:b</b> <b>Kernel: PBOC</b>	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 0- 终端仅支持“00”版本的 iDDA</td></tr> <tr> <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	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																																		
		预留																																		
C_TAG_TM_9F7B	<b>Tag: 9F7B</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: PBOC</b>	Ecash Floor limit If authorized amount higher than Ecash floor limit, transaction will request online authorization.																																		
C_TAG_TM_TRANS_LIMIT	<b>Tag: DF8124</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: PBOC</b>	Contactless Transaction Limit If authorized amount higher than Transaction limit, transaction will be terminated.																																		
C_TAG_TM_CVM_LIMIT	<b>Tag: DF8126</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: PBOC</b>	Contactless CVM Required Limit If authorized amount higher than CVM limit, transaction will be requested CVM.																																		
C_TAG_TM_FLOOR_LIMIT	<b>Tag: DF8123</b> <b>Length: 6</b>	Contactless Floor Limit If authorized amount higher than floor limit, transaction will																																		

	<b>Format:</b> BCD <b>Kernel:</b> PBOC	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: DF06</b> <b>Length: 2</b> <b>Format:b</b> <b>Kernel: VISA</b>	Reader configuration parameters Byte1: b8=Status Check b7=Amount,Authorized of Zero Check b6=Amount,Authorized of Zero Option (1b=Option1 0b=Option2, this bit is only applicable when the reader is online-capable)(Discover-Zero Amount Allowed Flag) b5=Reader Contactless Transaction Limit Check b4=Reader CVM Required Limit Check b3=Reader Contactless Floor Limit Check b2=Exception File b1=Certification Revocation List Byte2: b8-b1=RFU(0)
V_TAG_TM_9F66	<b>Tag: 9F66</b> <b>Length: 4</b> <b>Format:b</b> <b>Kernel: VISA</b>	TerminalTransaction Qualifiers Indicate terminal capabilities, requirements, and preferences to the card. Byte 1 bit 8: 1 = MSD supported bit 7: RFU (0) bit 6: 1 = qVSDC supported bit 5: 1 = EMV contact chip supported bit 4: 1 = Offline-only reader bit 3: 1 = Online PIN supported bit 2: 1 = Signature supported bit 1: 1 = Offline Data Authentication (ODA) for Online Authorizations supported. <i>Note:</i> Readers compliant to this specification set TTQ byte 1 bit 1 to 0b. Byte 2 bit 8: 1 = Online cryptogram required bit 7: 1 = CVM required bit 6: 1 = (Contact Chip) Offline PIN supported bits 5-1: RFU (00000) Byte 3 bit 8: 1 = Issuer Update Processing supported bit 7: 1 = Mobile functionality supported (Consumer Device CVM) bits 6-1: RFU (000000) Byte 4 RFU ('00')
V_TAG_TM_TRANS_LIMIT	<b>Tag: DF8124</b> <b>Length: 6</b> <b>Format:BCD</b>	Contactless Transaction Limit If authorized amount is <b>greater than or equal to</b> the Transaction limit, transaction will be terminated.

	<b>Kernel: VISA</b>	It is used in conjunction with <b>V_TAG_RD_RCP</b> (DF06).
<b>V_TAG_TM_CVM_LIMIT</b>	<b>Tag: DF8126</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: VISA</b>	Contactless CVM Required Limit If authorized amount is <b>greater than or equal to</b> the CVM limit, transaction will be requested CVM. It is used in conjunction with <b>V_TAG_RD_RCP</b> (DF06).
<b>V_TAG_TM_FLOOR_LIMIT</b>	<b>Tag: DF8123</b> <b>Length: 6</b> <b>Format:BCD</b> <b>Kernel: VISA</b>	Contactless Floor Limit If authorized amount is <b>greater than</b> the floor limit, transaction will be requested online authorization. It is used in conjunction with <b>V_TAG_RD_RCP</b> (DF06).  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).

#### 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   b7   b6   b5   b4   b3   b2   b1  </td><td>Specification 3.1</td></tr><tr><td>  ↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓  </td><td></td></tr><tr><td>  0   0       0  </td><td>Expresspay 1.0</td></tr><tr><td>  0   0       1  </td><td>RFU</td></tr><tr><td>  0   1       0  </td><td>Expresspay 2.0 Magstripe Only, or Expresspay ≥3.0 Magstripe-CVM Not Required</td></tr><tr><td>  0   1       1  </td><td>Expresspay ≥3.0 Magstripe-CVM Required</td></tr><tr><td>  1   0       0  </td><td>Expresspay 2.0 - EMV and Magstripe</td></tr><tr><td>  1   0       1  </td><td>RFU</td></tr><tr><td>  1   1       0  </td><td>Expresspay ≥3.0 EMV and Magstripe-CVM Not Required</td></tr><tr><td>  1   1       1  </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><u>A_TAG_PREAGAIN</u></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><u>A_TAG_TM_IN_CARD_BIN_RANG</u></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

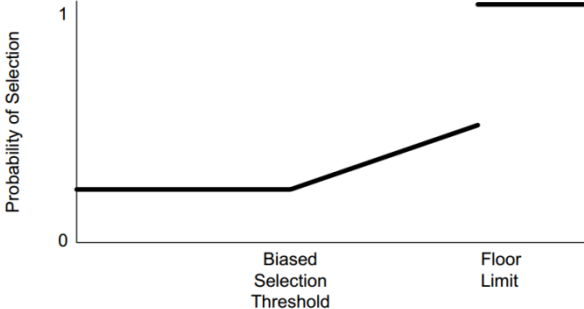
Kernel ID	EMV_KERNELID_DISCOVER	
Tag Name	Definition	Description
D_TAG_TM_RD_RCP	<b>Tag:</b> DF06 <b>Length:</b> 2 <b>Format:</b> b <b>Kernel:</b> <b>DISCOVER</b>	<p>Reader Configuration Parameters</p> <p>Byte1: DF Descriptions (1b=Enabled/Present 0b=Disabled/Not Present)  b8==Status Check(Value: 1-Support 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)</p> <p>Byte2:  b8=='Status Check' flag is present or not  (Flag: 1-Present 0-Not Present), using it conjunction with B1b8  b7-b1==RFU</p>
D_TAG_TM_9F66	<b>Tag:</b> 9F66 <b>Length:</b> 4 <b>Format:</b> b <b>Kernel:</b> <b>DISCOVER</b>	<p>TerminalTransaction Qualifiers</p> <p>Indicate terminal capabilities, requirements, and preferences to the card.</p> <p>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> <p>Byte2:  b8==Online Cryptogram required  b7==CVM Required  b6==(Contact chip) Offline PIN supported  b5-b1==RFU</p> <p>Byte3:  b8==Issuer Update Processing supported  b7==Consumer Device CVM supported  b6-b1==RFU</p> <p>Byte4:  b8-b1==RFU</p>
D_TAG_TM_TRANS_LIMIT	<b>Tag:</b> DF8124 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> <b>DISCOVER</b>	<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>
D_TAG_TM_FLOOR_LIMIT	<b>Tag:</b> DF8123	Terminal Contactless Floor Limit

	<b>Length:6</b> <b>Format:BCD</b> <b>Kernel:</b> <b>DISCOVER</b>	<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>	<b>Tag: DF8126</b> <b>Length:6</b> <b>Format:BCD</b> <b>Kernel:</b> <b>DISCOVER</b>	<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

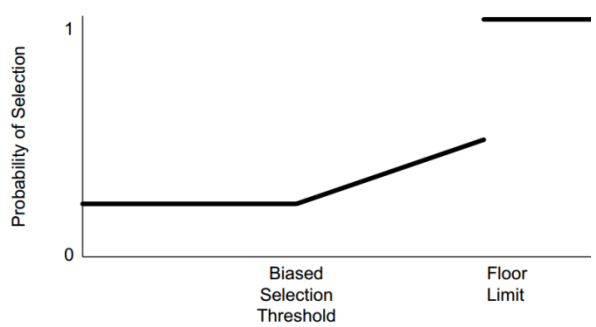
Kernel ID	EMV_KERNELID_JCB									
Tag Name	Definition	Description								
DEF_TAG_J_COMB_OPTION	Tag: DF918404  Length:2  Format:b  Kernel: JCB	Reader Configuration Parameters								
		Combination Options Byte 1 (Leftmost)								
		b8	b7	b6	b5	b4	b3	b2	b1	Meaning
		0								RFU
			1							Status Check 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>		
DEF_TAG_J_TIP	Tag: DF918408  Length:3  Format:b  Kernel: JCB	Terminal Interchange Profile								
		TIP Byte 1 (Leftmost)								
		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		
DEF_TAG_J_TRANS_LIMIT	Tag: DF918402  Length:6  Format:BCD  Kernel: JCB	Terminal Contactless Transaction Limit								
		If the Amount, Authorized is equal to or exceeds the Reader Contactless Transaction Limit, the transaction shall be terminate.								
DEF_TAG_J_FLOOR_LIMIT	Tag: DF918401  Length:6  Format:BCD  Kernel: JCB	Terminal Contactless Floor Limit								
		If authorized amount greater than floor limit, transaction will be requested online authorizat.								



DEF_TAG_J_CVM_LIMIT	<b>Tag: DF918403</b> <b>Length:6</b> <b>Format:BCD</b> <b>Kernel: JCB</b>	Terminal CVM Required Limit If the Amount, Authorized is equal to or exceeds the Reader CVM Required Limit, then CVM processing is required.
DEF_TAG_J_RS_MAX_PERCENT	<b>Tag: DF918405</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	Maximum Target Percentage (0-99), larger than Target Percentages.
DEF_TAG_J_RS_TARGET_PERCENT	<b>Tag: DF918406</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	Target Percentages(0-99)
DEF_TAG_J_RS_THRESH_VALUE	<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>
DEF_TAG_J_ONLINE_TWOPRE	<b>Tag: DF918410</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: JCB</b>	<p>联机发卡行更新两种方式判断:</p> <ol style="list-style-type: none"> <li>1、present And hold card (保存持卡)</li> <li>2、two present card(重新挥卡)</li> </ol> <p>通过判断联机回调函数交易数据 EMV_tTransData, 判断标签 DEF_TAG_J_ONLINE_TWOPRE 是否存在标签列表 auTLVData 中, 存在则执行重新挥卡操作, 否则保持持卡。</p>

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

Kernel ID	EMV_KERNELID_DEFINE	
Tag Name	Definition	Description
DEF_TAG_PSE_FLAG	<b>Tag:</b> DF918101 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	Application selection way: 0 - PSE selection first and then AID selection 1 - Only PSE selection 2 - Only AID selection 3 - Only PPSE selection 4 - PPSE First, AID selection Second (Discover ZIP Mode)
DEF_TAG_GAC_CONTROL	<b>Tag:</b> DF918102 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	Generate AC control flag 0 - Normal 1 - Force offline 2 - Force online 3 - Force decline
DEF_TAG_QUERY_ICCLOG	<b>Tag:</b> DF918103 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	Indicate if start aICC log query transaction: 0 - No 1 - Yes
DEF_TAG_SERVICE_TYPE	<b>Tag:</b> DF918104 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	Service Type, Refer to: <a href="#">Macro Definition</a>
DEF_TAG_START_RECOVERY	<b>Tag:</b> DF918105 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> PBOC	Indicate if start a recovery transaction for a torn transaction: 0 –Normal transaction process 1 - Current torn recovery process 2 – All torn recovery process
DEF_TAG_PAN_IN_BLACK	<b>Tag:</b> DF918106 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	Indicate if the PAN listed in exception file: 0 - No 1 - Yes
DEF_TAG_ACCUMULATE_AMOUNT	<b>Tag:</b> DF918107 <b>Length:</b> 6 <b>Format:</b> BCD <b>Kernel:</b> All	Serial offline approved amount accumulate for the same PAN.
DEF_TAG_CHV_STATUS	<b>Tag:</b> DF918108 <b>Length:</b> 1 <b>Format:</b> b <b>Kernel:</b> All	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
DEF_TAG_ONLINE_STATUS	<b>Tag:</b> DF918109 <b>Length:</b> 1 <b>Format:</b> b	Indicate the status of online communication: 0 – online success 1 – online failed

	Kernel: All	
DEF_TAG_AUTHORIZE_FLAG	Tag: DF91810A Length:1 Format:b Kernel: All	Indicate the result online authorization: 0 – Declined 1 – Approved
DEF_TAG_HOST_TLVDATA	Tag: DF91810B Length:Var Format:b Kernel: All	Hostresponse data while online authorization, such as issuer script, which is TLV format.
DEF_TAG_RAND_SLT_THRESHOLD	Tag: DF91810C Length:6 Format:BCD Kernel: All	<p>BiasedSelectionThreshold, if authorized amount higher thanBiasedSelectionThreshold, 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	Tag: DF91810D Length:1 Format:b Kernel: All	Target Percentages(0-99)
DEF_TAG_RAND_SLT_MAXPER	Tag: DF91810E Length:1 Format:b Kernel: All	Maximum Target Percentage (0-99), larger than Target Percentages.
DEF_TAG_TAC_DEFAULT	Tag: DF918110 Length:5 Format:b Kernel: All	Terminal Action Code, For Default
DEF_TAG_TAC_DECLINE	Tag: DF918111 Length:5 Format:b Kernel: All	Terminal Action Code, For Denial
DEF_TAG_TAC_ONLINE	Tag: DF918112 Length:5 Format:b	Terminal Action Code, For Online

	Kernel: All															
DEF_TAG_BALANCE_BEf_GAC	Tag: DF918113 Length:6 Format:BCD Kernel: PBOC\MASTER	ICC balance before Generate AC command.														
DEF_TAG_BALANCE_AFT_GAC	Tag: DF918114 Length:6 Format:BCD Kernel: PBOC\MASTER	ICC balance After Generate AC command.														
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_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 whenthe transaction amount is greater than the Reader CVMRequired 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 whenthe transaction amount is less than or equal to the ReaderCVM 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	<p>Mag-stripe CVM Capability – CVM Required</p> <p>Indicates the CVM capability of the termina in thecase of a mag-stripe mode transaction when the Amount authorized (Numeric) is greater than the Reader CVMRequired Limit</p> <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></td><td>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_NOCV M	Tag: DF918208 Length: Format: Kernel: MASTER	<p>Mag-stripe CVM Capability – No CVM Required</p> <p>Indicates the CVM capability of the Terminal in thecase of a mag-stripe mode transaction when the Amount authorized (Numeric) is less than or equal to the Reader CVMRequired Limit.</p> <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></td><td>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	<p>Message Hold Time</p> <p>Indicates the default delay for the processing of the next MSGsignal. The Message Hold Time is an integer in units of 100ms.</p>														
DEF_TAG_M_RF_HOLDTIME	Tag: DF91820A Length:1 Format:b Kernel: MASTER	<p>Hold Time Value</p> <p>Indicates the time that the field is to be turned off after thetransaction is completed if requested to do so by thecardholder device. The Hold Time Value is in units of 100ms.</p>														

DEF_TAG_D_ISSUERSCRIPT_EXCUTIVE	<b>Tag: DF918215</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: DISCOVER</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.
DEF_TAG_RESELECT_CONDITION	<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-最终选择卡片返回状态码非 9000 b7-最终选择卡片返回数据, 但未包含 PDOL(9F38) b6-最终选择卡片返回 PDOL, 但是 PDOL 未请求 TTQ(9F66) Byte2-Byte5 RFU
DEF_TAG_PPSE_6A82_TURNTO_AIDLIST	<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
DEF_TAG_CHECK_CAPK_INDEXLIST	<b>Tag: DF928104</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: PBOC/VISA</b>	qPBOC 和 qVSDC 卡片请求 TC 的情况下, 在读最后一条记录之前检查公钥索引列表 0x00-不检查 0x01-检查
DEF_TAG_CTL_AS_CB_FLAG	<b>Tag: DF928105</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	非接交易, 应用选择回调是否执行标志 0x00-不执行 0x01-执行
DEF_TAG_OBTAIN_FLAG	<b>Tag: DF928101</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	表示在某阶段执行结束后是否要输出执行 OBTAIN 回调 0x00-都不需要 0x01-脱机数据认证结束要执行
DEF_TAG_OBTAIN_RETURN_DATA	<b>Tag: DF928102</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	脱机数据认证结束, Obtain 回调输出数据 0x00-脱机数据认证成功 0x01-脱机数据认证失败
DEF_TAG_ONLIE_ODA_FAIL_FLOW_TYPE	<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
DEF_TAG_RESULT_CODE	<b>Tag: DF91810F</b> <b>Length:4</b> <b>Format:b</b> <b>Kernel: All</b>	交易返回码附加定位码, 具体用法请参考附录 C
DEF_TAG_ALLOW_DUP_ICC_SAMEVALUE	<b>Tag: DF918140</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	用于标识是否允许部分 IC 卡数据元重复出现, 且值相同: 0- 不允许重复 1- 允许重复 (以下几个数据元除外: 5A, 57,5F24,9F07,9F32,8C, 来自 PBOC 检测要求)

<b>DEF_TAG_ERROR_TYPE</b>	<b>Tag: DF91815A</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	用于标识终止交易时是何种错误信息提示 0x01-Try Another Interface 0x02-Try Another Payment 0x03-Use Another Card
<b>DEF_TAG_TRYGAIN_SPECIAL_RE TCODE</b>	<b>Tag: DF918162</b> <b>Length:1</b> <b>Format:b</b> <b>Kernel: All</b>	Indicate Try Again using Special Return Code, not using WaitCard(Flag) Event Callback Function 0x01: Using Special Return Code 0x00 or not set: Using onWaitCard(Flag) Envent, Flag= <b>EMV_FLAG_EXECUTE_CDCVM</b>
<b>DEF_TAG_FORCE_ONLINE_ALL</b>	<b>Tag: DF91815C</b> <b>Length:1</b> <b>Format: b</b> <b>Kernel: All</b>	用于指示内核是否对当前交易强制请求联机授权： 0 – 不强制请求联机授权，按正常流程走 1 –强制请求联机授权，同时 TVR 第 4 字节第 4 位即“商户 要求联机交易”位将会置位。

## A.9 TAG of EMV\_KERNELID\_RUPAY

Kernel ID	EMV_KERNELID_DISCOVER											
Tag Name	Definition	Description										
R_TAG_TM_TRANS_LIMIT	Tag: DF4C Length:6 Format:BCD Kernel: RUPAY	Contactless Transaction Limit  If the maximum transaction amount is <b>greater than or equal to (&gt;=)</b> the Contactless Transaction Limit, the Terminal must disable the contactless interface and only allow the transaction to proceed on the contact interface; if the Terminal does not have a contact interface, then the Terminal must reject the transaction.										
R_TAG_TM_FLOOR_LIMIT	Tag: DF8123 Length:6 Format:BCD Kernel: RUPAY	Terminal Contactless Floor Limit  If authorized amount is <b>exceeds(&gt;)</b> floor limit, transaction will be requested online authorizat <sup>on</sup> . Kernel will use Terminal Floor Limit(9F1B) when DF8123 is not present (if DF8123 and 9F1B are both present, then Kernel will use DF8123 in priority as Floor limit only).										
R_TAG_TM_CVM_LIMIT	Tag: DF4D Length:6 Format:BCD Kernel: RUPAY	Terminal CVM Limit  If a Contactless transaction amount <b>exceeds</b> the Terminal CVM Limit, the Terminal must exclude the ‘No CVM’ method from its list of supported CVMs.										
R_TAG_TM_SRTRANS_CAPP	Tag: DF8140 Length:2 Format:b Kernel: RUPAY	Terminal Service transaction capabilities  Byte 1    b8-b1=RFU Byte 2    b8-b7=RFU  b6=support Transit Transaction b5=support CAT3 Transaction b4=support Money Add Transaction b3=support Void Transaction  b2-b1=RFU										
R_TAG_TM_TIMELIMIT	Tag: DF8142 Length:1 Format:b Kernel: RUPAY	Recover AC Time limit, 0x1E=30 decimal seconds  Time limit is depend on different payment Environment, show bellow. <table><tr><th>Time limit</th><th>Description</th></tr><tr><td>30 seconds</td><td>Retail environment</td></tr><tr><td>5 seconds</td><td>Transit for rail</td></tr><tr><td>30 seconds</td><td>Transit for Bus (attended terminal)</td></tr><tr><td>5 seconds</td><td>Transit for Bus (unattended terminal)</td></tr></table>	Time limit	Description	30 seconds	Retail environment	5 seconds	Transit for rail	30 seconds	Transit for Bus (attended terminal)	5 seconds	Transit for Bus (unattended terminal)
Time limit	Description											
30 seconds	Retail environment											
5 seconds	Transit for rail											
30 seconds	Transit for Bus (attended terminal)											
5 seconds	Transit for Bus (unattended terminal)											
R_TAG_TM_DF16	Tag: DF16 Length:2 Format:b Kernel: RUPAY	Service ID - It is included only if the Terminal supports Service processing. If the Terminal initiates a Service-based transaction, then Service ID shall have a non-zero value.										
R_TAG_TM_DF3A	Tag: DF3A Length:2 Format:b Kernel: RUPAY	This specification introduces Additional Terminal Capabilities Extension (Tag DF3A), which shall contain all the proprietary qSPARC information to indicate the Service processing capabilities of a Non-Legacy Terminal. This data element shall only be returned by a										



		<p>Non-Legacy Terminal to a Non-Legacy Card if the Card specifies Tag DF3A in the PDOL.</p> <p>A Legacy Terminal shall return zeros for Tag DF3A, if the Card specifies it in the PDOL.</p> <table><tr><th>Tag</th><th>Length</th><th>Byte</th><th>Bit</th><th>Meaning</th></tr><tr><td rowspan="7">DF3A</td><td rowspan="7">5</td><td>1</td><td>8-1</td><td>RFU</td></tr><tr><td rowspan="3">2</td><td>8</td><td>RFU</td></tr><tr><td>7</td><td>1b - Service Creation supported</td></tr><tr><td>6-1</td><td>RFU</td></tr><tr><td>3</td><td>8-1</td><td>RFU</td></tr><tr><td>4</td><td>8-1</td><td>RFU</td></tr><tr><td>5</td><td>8-1</td><td>RFU</td></tr></table> <p>Table 97 - Additional Terminal Capabilities Extension (DF3A)</p>	Tag	Length	Byte	Bit	Meaning	DF3A	5	1	8-1	RFU	2	8	RFU	7	1b - Service Creation supported	6-1	RFU	3	8-1	RFU	4	8-1	RFU	5	8-1	RFU
Tag	Length	Byte	Bit	Meaning																								
DF3A	5	1	8-1	RFU																								
		2	8	RFU																								
			7	1b - Service Creation supported																								
			6-1	RFU																								
		3	8-1	RFU																								
		4	8-1	RFU																								
		5	8-1	RFU																								
R_TAG_TM_TSRQ	<p>Tag: DF8141</p> <p>Length:5</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>Service Qualifier, obtain from Acquirer</p> <table><tr><th>Tag</th><th>Length</th><th>Byte</th><th>Bit</th><th>Description</th></tr><tr><td>NA</td><td rowspan="4">5</td><td rowspan="2">1</td><td>8-2</td><td>Priority Number</td></tr><tr><td></td><td>1</td><td>0b - Cardholder Confirmation is needed 1b - Cardholder confirmation NOT needed</td></tr><tr><td></td><td>2-3</td><td>-</td><td>Service ID</td></tr><tr><td></td><td>4-5</td><td>-</td><td>Default Service Management Info</td></tr></table> <p>Table 123 - Service Qualifier</p>	Tag	Length	Byte	Bit	Description	NA	5	1	8-2	Priority Number		1	0b - Cardholder Confirmation is needed 1b - Cardholder confirmation NOT needed		2-3	-	Service ID		4-5	-	Default Service Management Info					
Tag	Length	Byte	Bit	Description																								
NA	5	1	8-2	Priority Number																								
			1	0b - Cardholder Confirmation is needed 1b - Cardholder confirmation NOT needed																								
		2-3	-	Service ID																								
		4-5	-	Default Service Management Info																								
R_TAG_TM_DF45	<p>Tag: DF45</p> <p>Length:Var.</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>Service Terminal Data</p> <p>Var (max 96)</p>																										
R_TAG_IC_DF47	<p>Tag: DF47</p> <p>Length:16</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>PRMiss, Service Key Component - Issuer</p> <p>This is one of the three components used in Service verification. Issuer plants this key in personalization and shares this with Acquirer. This key is used if Acquirer is creating a Service on the card. PRMiss (Tag DF47) shall be a 16 Bytes AES key, generated and injected onto the Card at the time of personalization by the Issuer. This data is present for card with AVN &lt; 0002 - spec v1.0 legacy cards. PRMiss is sent by card in spec 2.0 for non-legacy cards.</p>																										
R_TAG_TM_DF48	<p>Tag: DF48</p> <p>Length:8</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>PRMacq, Service Key Component - Acquirer</p> <p>This is one of the three components used in Service verification. Card uses this key to verify terminal's Service data write. PRMacq (Tag DF48) shall be an 8-byte key generated by the Acquirer.</p>																										
R_TAG_IC_KCV	<p>Tag: DF54</p> <p>Length:3</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>PRMacq KCV - This is Key Check Value of PRMacq (DF48)</p> <p>It is a 3-digit key check value generated on Acquirer key i.e. PRMacq (Tag DF48) and verified by Card on Service creation.</p>																										
R_TAG_IC_PRMACQ_KEYINDEX	<p>Tag: DF4E</p> <p>Length:1</p> <p>Format:b</p> <p>Kernel: RUPAY</p>	<p>PRMacq Key Index</p> <p>The PRMacq key is used by the Terminal and the Card to create and update a Service. The Terminal provides the value of the PRMacq key index during Service Creation (key planting).</p>																										
R_TAG_TM_ONLINE_ORNOT	<p>Tag: DF8144</p> <p>Length:1</p>	<p>Indicates an Actual go online</p> <p>0-Actual not go online(Unable go Online)</p>																										

	<b>Format:</b> b <b>Kernel:</b> RUPAY	1-Actual go online																																																							
<b>R_TAG_IC_DF07</b>	<b>Tag:</b> DF07 <b>Length:</b> Var. <b>Format:</b> b <b>Kernel:</b> RUPAY	<p>Service Directory(SrDIR)</p> <p>This includes the list of the services available on the card.</p> <table><tr><th>Tag</th><th>Length</th><th>Byte</th><th>Bit(s)</th><th>Description</th></tr><tr><td rowspan="12">DF07</td><td rowspan="12">Var.</td><td>1</td><td>-</td><td>Service Version (SVER)</td></tr><tr><td>2</td><td></td><td>RFU</td></tr><tr><td>3-12</td><td>-</td><td>Service Label (PAN+PSN)</td></tr><tr><td>13</td><td>-</td><td>Service Limit (SLIM)</td></tr><tr><td>14</td><td>-</td><td>Service Counter (SCON)</td></tr><tr><td rowspan="2">15</td><td>8-5</td><td>0000b</td></tr><tr><td>4-1</td><td>Service SFI</td></tr><tr><td>16</td><td>-</td><td>DF50 - Service Permanent Counter (SCPER)</td></tr><tr><td>17</td><td>-</td><td>DF12 - Service Permanent Limit(SPLIM)</td></tr><tr><td>18-21</td><td>-</td><td>Service 1 Folder (Service ID 2 bytes, Service Control 2 bytes)</td></tr><tr><td>22-25</td><td>-</td><td>Service 2 Folder (Service ID 2 bytes, Service Control 2 bytes)</td></tr><tr><td>Var.</td><td>-</td><td>Service Folder (see Table 13): Contains list of all Service IDs and their associated Service Control data currently active on the Card i.e. four bytes of information for each Service.</td></tr></table> <p>Table 12 - DF07 Service Directory</p> <table><tr><th>Tag</th><th>Length</th><th>Byte</th><th>Bit</th><th>Definition</th></tr><tr><td rowspan="2">DF32</td><td rowspan="2">Var.</td><td>1-2</td><td>-</td><td>Service ID (DF16)</td></tr><tr><td>3-4</td><td>-</td><td>Service Control (see Table 17)</td></tr></table> <p>Table 13 - DF32 Service Folder</p>	Tag	Length	Byte	Bit(s)	Description	DF07	Var.	1	-	Service Version (SVER)	2		RFU	3-12	-	Service Label (PAN+PSN)	13	-	Service Limit (SLIM)	14	-	Service Counter (SCON)	15	8-5	0000b	4-1	Service SFI	16	-	DF50 - Service Permanent Counter (SCPER)	17	-	DF12 - Service Permanent Limit(SPLIM)	18-21	-	Service 1 Folder (Service ID 2 bytes, Service Control 2 bytes)	22-25	-	Service 2 Folder (Service ID 2 bytes, Service Control 2 bytes)	Var.	-	Service Folder (see Table 13): Contains list of all Service IDs and their associated Service Control data currently active on the Card i.e. four bytes of information for each Service.	Tag	Length	Byte	Bit	Definition	DF32	Var.	1-2	-	Service ID (DF16)	3-4	-	Service Control (see Table 17)
Tag	Length	Byte	Bit(s)	Description																																																					
DF07	Var.	1	-	Service Version (SVER)																																																					
		2		RFU																																																					
		3-12	-	Service Label (PAN+PSN)																																																					
		13	-	Service Limit (SLIM)																																																					
		14	-	Service Counter (SCON)																																																					
		15	8-5	0000b																																																					
			4-1	Service SFI																																																					
		16	-	DF50 - Service Permanent Counter (SCPER)																																																					
		17	-	DF12 - Service Permanent Limit(SPLIM)																																																					
		18-21	-	Service 1 Folder (Service ID 2 bytes, Service Control 2 bytes)																																																					
		22-25	-	Service 2 Folder (Service ID 2 bytes, Service Control 2 bytes)																																																					
		Var.	-	Service Folder (see Table 13): Contains list of all Service IDs and their associated Service Control data currently active on the Card i.e. four bytes of information for each Service.																																																					
Tag	Length	Byte	Bit	Definition																																																					
DF32	Var.	1-2	-	Service ID (DF16)																																																					
		3-4	-	Service Control (see Table 17)																																																					

## 附录 B 宏定义集合

宏名称定义	值	注释
信号类型 (Signal Type)		
EMV_SIGNAL_ACT	0x01	从应用选择开始, 激活一笔新交易
EMV_SIGNAL_NEXT	0x02	回调正常返回, 执行下一步操作
EMV_SINGAL_ACT_GPO	0x11	从 GPO 开始, 激活一笔新交易
EMV_SIGNAL_STOP	0xF0	终止交易
EMV_SIGNAL_CLEAN	0xC0	清理内核交易数据
内核类型 (Kernel ID)		
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_RUPAY	0x0D	RUPAY
EMV_KERNELID_NSICC	0xDA	Indonesian
EMV_KERNELID_DEFINE	0xDE	Kernel Defined
交易流程标识 (Flow Type)		
EMV_FLOWTYPE_EMV	0x01	EMV/PBOC 标准借贷记
EMV_FLOWTYPE_ECASH	0x03	接触式小额支付 (电子现金)
EMV_FLOWTYPE_QPBOC	0x11	非接小额支付 (qPBOC)
EMV_FLOWTYPE_PBOC_CTLESS	0x12	非接标准借贷记
EMV_FLOWTYPE_MSD	0x13	MSD 交易
EMV_FLOWTYPE_MSD_LEGACY	0x14	MSD Legacy 交易
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

EMV_FLOWTYPE_R_LEGACY	0x61	RuPay EMV Legacy Mode
EMV_FLOWTYPE_R_NONLEGACY	0x62	RuPay EMV Non-Legacy Mode
服务类型 (Service Type)		
EMV_SERVETYPE_GOOD	0x00	商品
EMV_SERVETYPE_SERVICE	0x00	服务
EMV_SERVETYPE_CASH	0x01	现金
EMV_SERVETYPE_CASHBACK	0x09	返现
EMV_SERVETYPE_REFUND	0x20	退款
持卡人验证类型 (CVM Flag)		
EMV_CVMFLAG_NOCVM	0x00	无需 CVM 验证
EMV_CVMFLAG_OFFLINEPIN	0x01	脱机 PIN
EMV_CVMFLAG_ONLINEPIN	0x02	联机 PIN
EMV_CVMFLAG_SIGNATURE	0x03	签名
EMV_CVMFLAG_OLPIN_SIGN	0x04	联机 PIN 加签名
EMV_CVMFLAG_CDV	0x05	Consumer Device Verification(qVSDC/qPBOC)
EMV_CVMFLAG_CCV	0x06	Confirmation Code Verified(PayPass)
EMV_CVMFLAG_CERTIFICATE	0x11	证件验证(证件类型: 00-身份证 01-军官证 02-护照 03-入境证 04-临时身份证 05-其他)
EMV_CVMFLAG_ECASHPIN	0x21	电子现金充值 PIN
行为分析类型 (AC Type)		
EMV_ACTION_AAC	0x00	拒绝交易
EMV_ACTION_TC	0x01	批准交易
EMV_ACTION_ARQC	0x02	请求联机
动作标识 (Action Flag)		
EMV_FLAG_ADD	0x01	添加
EMV_FLAG_DELETE	0x02	删除
EMV_FLAG_CLEAR	0x03	清除所有
内核指令		
EMV_INS_SET_TORN	0xA1	内核发送闪卡记录, 应用端存储记录数据
EMV_INS_DEL_TORN	0xD1	通知应用端删除闪卡记录
EMV_INS_DISPLAY	0xA2	通知应用端显示信息, 信息格式参考结构体: <a href="#">EMV_tDisplayMsg</a>
EMV_INS_TLVDATA	0xA3	发送 TLV 数据串
EMV_INS_CLOSERF	0xA4	通知应用端关闭非接载波
EMV_INS_DBLOG	0xDB	发送 debug 日志信息
EMV_INS_GET_TORN	0xB1	内核请求闪卡记录
EMV_INS_APPSELECT_DATA	0xC1	发送应用选择数据
EMV_INS_SET_FAIL_WATER	0xA5	发生闪卡, 通知应用保存失败交易记录
EMV_INS_DEL_FAIL_WATER	0xA6	通知应用删除闪卡保存的失败交易记录
信息标识 (Message ID)		
EMV_MSGID_CARD_READ_OK	0x17	Read card finished
EMV_MSGID_TRY_AGAIN	0x21	Try again

EMV_MSGID_APPROVED	0x03	Transaction approved
EMV_MSGID_APPROVED_SIGN	0x1A	Transaction approved and requesting signature
EMV_MSGID_DECLINED	0x07	Transaction declined
EMV_MSGID_ERR_OTH_CARD	0x1C	Transaction error, please try other card.
EMV_MSGID_INSERT_CARD	0x1D	Please insert IC card
EMV_MSGID_SEE_PHONE	0x20	Please check cell phone.
EMV_MSGID_AUTH_WAIT	0x1B	Waiting authorization
EMV_MSGID_CLEAR_DISPLAY	0x1E	Clear screen display
EMV_MSGID_ICC_ACCOUNT	0x1F	Icc Account
EMV_MSGID_PCII	0xF1	Display message according PCII
EMV_MSGID_UNMATCH_PAN	0xF2	In current tron recovery process, unmatched pan
EMV_MSGID_READ_CARD_FAIL	0xF3	In All flash tron recovery process, read card fail
EMV_MSGID_ONLINE_ODA_RESULT	0xF4	In OnlineOda process, the result of ODA
错误定位码		
EMV_L1_ERR_TIMEOUT	0x01	ICC APDU communicate time out
EMV_L1_ERR_TRANSMISSION	0x02	ICC APDU transmission error
EMV_L1_ERR_PROTOCOL	0x03	ICC transmit protocol error
EMV_L2_ERR_ICC_DATA_MISS	0x01	ICC Data missing
EMV_L2_ERR_CAM_FAILED	0x02	CAM Fail
EMV_L2_ERR_ICC_STATUS	0x03	APDU Status Error
EMV_L2_ERR_PARSING	0x04	ICC data parsing error.
EMV_L2_ERR_MAX_EXCEEDED	0x05	Exceed max limit.
EMV_L2_ERR_ICC_DATA	0x06	ICC data error.
EMV_L2_ERR_MAG_NOT_SUP	0x07	Don't support magnetic
EMV_L2_ERR_NO_PPSE	0x08	Don't support PPSE
EMV_L2_ERR_PPSE_FAULT	0x09	PPSE fault
EMV_L2_ERR_NO_CAND_AID	0x0A	Candidate AID list is empty.
EMV_L2_ERR_TERM_DATA	0x0F	Terminate parameter error.
EMV_L3_ERR_TIMEOUT	0x01	Time out
EMV_L3_ERR_STOP	0x02	Transaction being terminated
EMV_L3_ERR_AMOUNT	0x03	Amount is absent
寻卡回调参数标识		
EMV_FLAG_NORMAL	0x00	正常交易的寻卡标志
EMV_FLAG_SHOW_CARD_AGAIN	0x01	重新激活卡片
EMV_FLAG_ISS_SCRIPT_UPDATE	0x02	提示再次出卡，但不显示金额
EMV_FLAG_EXECUTE_CDCVM	0x03	CDCVM 未执行，提示用户根据手机提示操作，并重新挥卡

## 附录 C 交易返回码

交易返回码是交易结束事件回调 API 的第一个出参 **uiResult** ,用于指示当前交易的运行状态和错误类型，如下所示：

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

交易返回码分为三大类：

### 1、交易正常结束

当前唯一标识交易正常结束的返回码：**EMV\_RESULT\_NORMAL**

### 2、常见错误返回码

- 编码由 2 个字节组成，编码格式为：0xEX XX
- 常见错误返回码具有通用，高频出现的特征，通常与卡组织类型无关，内核已经定义成宏，方便上层应用的识别。

### 3、非常见错误返回码

- 编码由 4 个字节组成，编码格式为： 0xFN XX YY ZZ
- 其中第四个字节 ZZ，如果值为 0xFX，那么要定位一个具体错误，还需要通过标签 DEF\_TAG\_RESULT\_CODE ( DF91810F ) 获取四个字节的辅助定位码。  
否则不需要通过标签 DEF\_TAG\_RESULT\_CODE 辅助定位。

编程指南：

```
void EXEP_vEndProcess(uint uiResult, const EMV_tTransData *ptTransData)
{
    Uint uiLen;
    Uchar auResultCode[10];
    switch(uiResult)
    {
        case EMV_RESULT_NORMAL:
            Log("交易正常结束，交易状态请查看第二个出参 EMV_tTransData");Break;
        case EMV_RESULT_NOAPP:
```

```

        Log("终端卡片应用未匹配, 无候选应用, 交易终止");Break;
case EMV_RESULT_NOPUBKEY:
    Log("非接小额支付, 检测到未设置交易公钥, 交易终止");Break;
case EMV_RESULT_EXPIRY:
    Log("卡片应用过期, 交易终止");Break;
case EMV_RESULT_STOP:
    Log("STOP 信号请求终止, 交易主动退出");Break;
case EMV_RESULT_REPOWERICC:
    Log("请重新对卡片上下电进行交易");Break;
case EMV_RESULT_REFUSESERVICE:
    Log("卡片不允许服务, 交易终止");Break;
case EMV_RESULT_CARDLOCK:
    Log("卡片锁定(SW=6A81)");Break;
case EMV_RESULT_APPLOCK:
    Log("应用锁定(SW=6283)");Break;
case EMV_RESULT_EXCEED_CTLMT:
    Log("交易金额超过非接限额, 交易终止");Break;
case EMV_RESULT_APDU_ERROR:
    Log("APDU 执行失败, 请重试");Break;
case EMV_RESULT_APDU_STATUS_ERROR:
    Log("APDU 状态码错误, 请检查卡片是否正常");Break;
default:
    uiLen=0;
    memset(auResultCode, 0, sizeof(auResultCode))
    if ((uiResult&0xF0)==0xF0)
        //需要通过标签 DEF_TAG_RESULT_CODE 获取辅助定位码
        EMV_Kern_uiGetTLV(tEMVObject,DEF_TAG_RESULT_CODE,4,auResultCode, &uiLen);
    }
    if(!uiLen)
    {    Log("其它错误, 交易终止, 交易返回码为:%04X",uiResult); }
    else
    {    Log("其它错误, 交易终止, 交易返回码为:%04X+%04X",uiResult, auResultCode); }
    break;
}
}

```

交易正常结束		
EMV_RESULT_NORMAL	0x0000	交易正常结束
常见错误返回码		
EMV_RESULT_BUSY	0xEE01	EMV 工作线程忙碌中, 请稍候再尝试
EMV_RESULT_NOAPP	0xEE02	终端卡片应用未匹配, 无候选应用

<b>EMV_RESULT_NOPUBKEY</b>	0xEE03	非接小额支付, 检测到未设置交易公钥
<b>EMV_RESULT_EXPIRY</b>	0xEE04	卡片应用过期, 交易终止
<b>EMV_RESULT_FLASHCARD</b>	0xEE06	非接快速支付, 发生闪卡
<b>EMV_RESULT_STOP</b>	0xEE07	STOP 信号请求终止, 交易主动退出
<b>EMV_RESULT_REPOWERICC</b>	0xEE08	卡片通讯错误
<b>EMV_RESULT_REFUSESERVICE</b>	0xEE09	卡片不允许服务
<b>EMV_RESULT_CARDLOCK</b>	0xEE0A	卡片锁定 (SW=6A81)
<b>EMV_RESULT_APPLOCK</b>	0xEE0B	应用锁定 (SW=6283)
<b>EMV_RESULT_EXCEED_CTLMT</b>	0xEE0C	交易金额超过非接限额
<b>EMV_RESULT_APDU_ERROR</b>	0xEE0D	APDU 执行失败
<b>EMV_RESULT_APDU_STATUS_ERROR</b>	0xEE0E	APDU 状态码错误
<b>EMV_RESULT_ALL_FLASH_CARD</b>	0xEE0F	返回全部闪卡待处理流程
<b>非常见错误返回码</b>		
<b>FN XX YY ZZ</b>	当 ZZ 的值>=0xF0 时, 需要获取辅助定位码 DF91810F	
<b>DEF_TAG_RESULT_CODE</b>	辅助定位码, Hex 编码, 四个字节。	



## 附录 E 常见问题解答

### 非接交易相关

#### A1. 如何走非接快速支付（qPBOC）脱机交易？

如下必要条件，请逐项检查：

1. 开始交易前，必须设置 **C\_TAG\_TM\_9F66**，  
推荐值：支持联机和脱机:26 00 00 80，仅脱机:2A 00 00 80
2. **C\_TAG\_TM\_9F66**：第 1 字节的第 7,8 位，置 0，第 6 位置 1.
3. 必须设置 **C\_TAG\_TM\_FLOOR\_LIMIT**
4. 交易金额必须小于或等于 **C\_TAG\_TM\_FLOOR\_LIMIT**
5. 卡片必须支持 qPBOC 并且卡片余额充足

#### A2. 如何走非接快速支付（qPBOC）联机交易？

如下必要条件，请逐项检查：

1. 必须设置 **C\_TAG\_TM\_9F66** 推荐值: 26 00 00 80
2. **C\_TAG\_TM\_9F66**：第 1 字节第 7,8,4 位，置 0，第 6 位置 1.
3. 必须设置 **C\_TAG\_TM\_FLOOR\_LIMIT**（强制联机，设置限额为 0）
4. 交易金额必须大于 **C\_TAG\_TM\_FLOOR\_LIMIT**

#### A3. 如何走非接标准借贷记（完整流程）交易？

如下必要条件，请逐项检查：

- 1、必须设置 **C\_TAG\_TM\_9F66** 推荐值: 46 00 00 80
- 2、**C\_TAG\_TM\_9F66**：第 1 字节第 7 位置 1，第 6 为置 0。

#### A4. 如何支持国密算法并且查看是否采用了国密算法？

支持国密设置 **C\_TAG\_TM\_DF69** 值为 1，内核处理逻辑为：如果卡片支持国密算法，则采用国密算法，如果卡片不支持国密算法，则采用 RSA 算法。应用判断内核最终是否有采用国密算法，可通过读记录后处理事件的输出参数 EMV\_tReadRecord-> ucAlgorithmID 来辨别。

#### A5. 如何获取非接卡片交易前余额？

参考交易流程图：[5.9 读卡片余额交易](#)

#### A6. 如何走非接免签免密脱机批准交易？

假设限额是 300 元，当交易金额小于等于 300 元时，免签免密；当超过 300 元时，请求签名或者联机 PIN，可做如下设置：

- 1、开始交易前，设置 **C\_TAG\_TM\_9F66**  
若超过 300 请求签名，设置值为：22 00 00 80

若超过 300 请求联机 PIN，设置值为：24 00 00 80

2、设置 **C\_TAG\_TM\_CVM\_LIMIT** 与 **C\_TAG\_TM\_FLOOR\_LIMIT**

若超过 300 请求签名，设置 **C\_TAG\_TM\_CVM\_LIMIT** 为 300 元，**C\_TAG\_TM\_FLOOR\_LIMIT** 超过 300 元

若超过 300 请求联机 PIN，设置 **C\_TAG\_TM\_CVM\_LIMIT** 与 **C\_TAG\_TM\_FLOOR\_LIMIT** 同时为 300 元

3、如果交易请求免签免密，交易金额必须小于或等于 **C\_TAG\_TM\_CVM\_LIMIT**，否则金额大于 **C\_TAG\_TM\_CVM\_LIMIT**

4、卡片必须支持 qPBOC 并且卡片余额充足

**A7. 为什么非接快速支付（qPBOC）走联机未收到请求联机 PIN 事件？**

非接快速支付交易联机不会产生联机 PIN 请求事件，而是在交易结束通知事件产生后，由输出参数 EMV\_tTransData->ucCVM 指定是否需要联机 PIN. 应用自行执行联机 PIN 验证操作，后续处理无需与内核交互。

参考交易流程图：[5.5 PBOC 非接快速支付交易 qPBOC](#)

**A8. 如何获取非接卡片主账号？**

读记录后处理事件产生后，通过事件出参或者通过标签 5A 获取。

**A9. 为什么非接交易刚激活就终止了？**

非接交易金额必须小于非接限额 **C\_TAG\_TM\_TRANS\_LIMIT**

## 接触交易相关

### B1. 为什么联机交易有时会请求联机 PIN 有时又没有请求？

如下必要条件，请逐项检查：

- 1、终端必须支持联机 PIN，检查终端性能 **EMV\_TAG\_TM\_CAP** 第 1 字节第 7 位是否置位。
- 2、检查卡片返回的 CVMList 标签 8E，是否第一条 CVM 验证方法为联机 PIN，即 42 03 或者 02 03。

针对国内联机必须输入联机 PIN 的需求，应用端应该判断这种情况，就是如果内核发送了联机处理事件，但是未请求联机 PIN 事件，应用允许强制输入联机 PIN，PIN block 不需要传给内核。

### B2. 如何设置强制联机？

设置标签 **DEF\_TAG\_GAC\_CONTROL** 值为 02，详细参考[标签定义](#)

### B3. 如何走电子现金交易脱机批准交易？

如下必要条件，请逐项检查：

- 1、标签 **C\_TAG\_TM\_9F7A** 设置为 1
- 2、**EMV\_TAG\_TM\_TERMTYPE** 必须支持脱机，详细参考[标签定义](#)
- 3、卡片必须是电子现金卡并且卡片余额充足。

### B4. 如何支持国密算法并且查看是否采用了国密算法？

支持国密设置 **C\_TAG\_TM\_DF69** 值为 1，内核处理逻辑为：如果卡片支持国密算法，则采用国密算法，如果卡片不支持国密算法，则采用 RSA 算法。应用判断内核最终是否有采用国密算法，可通过**读记录后处理**事件的输出参数 **EMV\_tReadRecord->ucAlgorithmID** 来辨别。

### B5. 如何获取卡片主账号？

EMV 交易的卡片主账号只能在**读记录后处理**事件产生获得，应用可通过事件输出参数 **EMV\_tReadRecord->auPAN** 来获取，或者通过取标签值 5A（持卡人主账号）或者 57（二磁道等效数据）获取。

### B6. 如何获取电子现金卡片余额？

参考交易流程图：[5.9 读卡片余额交易](#)

### B7. 如何查看脚本执行结果？

脚本执行结果产生的前提是交易请求了联机，并且联机返回数据中带有 71 或者 72 标签的脚本，联机返回数据格式必须是 TLV 数据串。应用在**交易结束通知**事件产生后获得，可通过事件输出参数 **EMV\_tTransData->auScriptResult** 来获取，或者通过取标签值 DF31 获取。

### B8. 如何判断当前卡片是否支持电子现金？

在**最终选择后处理**事件产生后，通过检查标签 9F38（PDOL）是否包含 9F7A01 来鉴别。如果有包含 9F7A01 则支持，否则不支持。

### B9. 如何判断当前交易是否是电子现金交易？

在读记录后处理事件产生后，通过查看输出参数 EMV\_tReadRecord->ucFlowType 获知。如果值为 **EMV\_FLOWTYPE\_ECASH** 则说明是电子现金交易。

### B10. 脱机 PIN 验证失败如何查找问题？

由于脱机 PIN 验证内核是调用应用注册的验证回调函数直接去验证的，所以如果脱机 PIN 验证失败，应用应该先检查回调函数调用底层验证 PIN 接口的返回值是否是成功的，然后在检查返回的 SW12 是否是 9000，根据返回值以及 SW12 去查找对应接口的说明文档，确定问题所在！如果驱动返回验证返回值是 **0xe6** 或者 **0xe7**，应用应该转换成 SUCCESS（0x00）给内核，这样内核才会继续去判断 SW12。

### B11. EMV\_TAG\_TM\_TRANSTYPE（9C）和 DEF\_TAG\_SERVICE\_TYPE（DF918104）如何使用

- 1、9C 目前只定义 0-商品和服务，其它交易类型值可能因收单系统不同而不同，因此如果明确是商品和服务类型的交易（例如消费），只需设置 9C 等于 0 即可，不需要设置 DEF\_TAG\_SERVICE\_TYPE。
- 2、对于非商品和非服务类型的交易，需要设置 DEF\_TAG\_SERVICE\_TYPE（DF918104），其值的范围定义参照开发指南：

<b>EMV_SERVETYPE_GOOD</b>	0x00	商品
<b>EMV_SERVETYPE_SERVICE</b>	0x00	服务
<b>EMV_SERVETYPE_CASH</b>	0x01	现金
<b>EMV_SERVETYPE_CASHBACK</b>	0x09	返现
<b>EMV_SERVETYPE_REFUND</b>	0x20	退款

内核会根据 DEF\_TAG\_SERVICE\_TYPE 的不同的值做不同的处理。然而对于 9C，以各自收单系统定义的值为准，因为 9C 需要上送后台进行校验。

给大家推荐一个比较简单保险的处理方法：

- 1、DEF\_TAG\_SERVICE\_TYPE（DF918104）设置为 0（非现金和返现类交易）
- 2、EMV\_TAG\_TM\_TRANSTYPE（9C）根据当前交易类型按收单系统要求的值进行设置，如果不清楚就设置为 0。