ExpressPay 2.0: Application migration

ICO-OPE-00136-V2-EN

23 May 2012

Contents

1. Introduction

The ExpressPay 1 kernel supports Magstripe transactions only.
The new ExpressPay 2 kernel allows performing both Magstripe and EMV transactions.

This note explains how an existing application must be modified to perform a Magstripe transaction using ExpressPay 2 kernel as it did with the ExpressPay 1 kernel.

The changes could be summarised as follow:
- New tags
- New value returned by the ExpressPay_DoTransaction() function
- New behavior

2. New tags

Some new tags must be added in the parameter file.

2.1. List of application version number supported by the terminal

TAG_EXPRESSPAY_INT_TERMINAL_AVN_LIST (0x9F91820A)

This consists of a concatenation of AVN, each of them is coded on 2 bytes, up to 128 bytes (i.e. up to 64 application version numbers).

To perform a Magstripe transaction using ExpressPay 2 kernel, the list must contain the version 0x0001.

2.2. ExpressPay Terminal Capabilities

TAG_EXPRESSPAY_TERMINAL_CAPABILITIES (0x9F6D)

The Terminal Type tag used by the kernel when performing a transaction is a modified terminal type (including the two first bits of the TAG_EXPRESSPAY_TERMINAL_CAPABILITIES tag).

Here is the coding of the terminal type tag:

Terminal Type - EMV Tag '9F35'

Terminal Type								
B8	B7	B6	B5	B4	B3	B2	B1	Meaning
		0	1					Financial Institution
		1	0					Merchant
		1	1					Cardholder
					0	0	1	Attended – Online Only
					0	1	0	Attended – Offline with Online Capability
					0	1	1	Attended – Offline Only
					1	0	0	Unattended – Online Only
					1	0	1	Unattended – Offline with Online Capability
					1	1	0	Unattended – Offline Only

Here is the coding of the ExpressPay Terminal Capabilities tag:

Expresspay Terminal Capabilities '9F6D'

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
0	0			0				Expresspay 1.0
0	1			0				Expresspay 2.0 – Magstripe Only
1	0			0				Expresspay 2.0 – EMV and Magstripe

Note: Bits 6 and 5 and Bits 3 to 1 are reserved and must be set to zero. These bits will correspond to the values defined in Terminal Type {EMV} '9F35'.

To perform a Magstripe transaction using ExpressPay 2 kernel, the ExpressPay Terminal Capabilities tag must be set to 0x40.

The Terminal Type tag (0x9F35) must be modified to contain the ExpressPay Terminal Capabilities (0x9F6D) as shown by the following table:

Terminal Type – Modified

As returned from an Expresspay capable Terminal – consisting of EMV Terminal Type '9F35' (Table 6) OR'd with Expresspay Terminal Capabilities '9F6D' (Table 7)

Terminal Type

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
		0	1					Financial Institution
		1	0					Merchant
		1	1					Cardholder
					0	0	1	Attended – Online Only
					0	1	0	Attended – Offline with Online Capability
					0	1	1	Attended – Offline Only
					1	0	0	Unattended – Online Only
					1	0	1	Unattended – Offline with Online Capability
					1	1	0	Unattended – Offline Only
0	0			0				Expresspay 1.0
0	1			0				Expresspay 2.0 – Magstripe Only
1	0			0				Expresspay 2.0 – EMV and Magstripe

American Express issued the "Specification Bulletin 01" applicable immediately to the ExpressPay 2.0.1 and 2.0.2 Terminal Specifications.
This bulletin clarifies the usage of the Terminal Type data. This data is modified during the transaction process, but only the unmodified data must be sent to the acquirer host.

The application has to send to the acquirer host either the original Terminal Type that it sent to the kernel or the value from the kernel with a mask applied to clear the modifications (*TerminalType* & 0x37).

Example extracted from the specification bulletin:

IF Terminal Type (Tag 9F35) in Terminal Configuration data	= '22'
AND Expresspay Terminal Capabilities (Tag 9F6D)	= '80'
THEN Terminal Type – Modified	= 'A2'

In the above example the value of the Terminal Type – Modified that is provided to the Card in the GET PROCESSING OPTIONS command would be 'A2', however the value of the Terminal Type (9F35) that would be sent in any authorization or submission messages to an acquirer would remain as '22'.

3. New values returned by ExpressPay_DoTransaction() function

3.1. Response to Final Select command

If the status word of the response to the Final Select is not 0x9000, the ExpressPay kernel will return one of the following results:
- KERNEL_STATUS_CARD_BLOCKED (0x6A81)
- KERNEL_STATUS_APPLICATION_BLOCKED (0x6283)
- KERNEL_STATUS_REMOVE_AID (else)

If one of these results is received, **the application must select the next AID in the candidate list** instead of terminating the transaction.

3.2. Online Authorization

In the case of an online authorization, the ExpressPay_DoTransaction function now returns EXPRESSPAY_STATUS_MAGSTRIPE_ONLINE_AUTHORISATION (0x0207) (defined in ExpressPay_API.h file) instead of returning KERNEL_STATUS_ONLINE_AUTHORISATION.

4. New behavior

4.1. Online Only terminals

With an Online only terminal which is only capable to approve transactions online, if the transaction is unable to go online then **the transaction must be declined by the terminal**.

4.2. Online authorization failure + SDA

Specification 2.0.2, section 14.2 mandates that:
- if SDA is to be performed and
- "if the card responds with a need to go online" and
- "if the transaction is being processed in Magstripe Mode or in a Terminal capable of only processing Partial Online Transactions then SDA is not performed" but "if the transaction is unable to go online then **the transaction is declined by the terminal**".

To check if SDA is the selected method, the custom application must get the *TAG_EXPRESSPAY_INT_ODA_METHOD* tag from the ExpressPay kernel and then check that its value is *EXPRESSPAY_ODA_METHOD_SDA*.