ExpressPay 3.0: Application migration

ICO-OPE-00209-V1-EN

4 July 2012

Ingenico – 28/32 boulevard de Grenelle – 75015 Paris – France
http://www.ingenico.com

Contents

1. Introduction

The new ExpressPay 3 kernel is compliant with the ExpressPay 3.0 Terminal Specification (February 2012) and the additional following bulletins:

- Bulletin 04 - Terminal Type Capabilities.
- Bulletin 05 - Application Selection Priority.
- Bulletin 06 - Mobile Passcode Processing Requirements.

ExpressPay 3 provides two new major changes:

- Mobile payment.
- Issuer Scripts acceptance.

These new functionalities will not be discussed here. If you need help to know how to modify your application to support these new features, please read the ExpressPay kernel documentation from the *Easy Path to C'Less* package.

This note explains how an existing application must be modified to perform an ExpressPay transaction (in Magstripe or EMV mode) using the ExpressPay 3 kernel as it did with the ExpressPay 2 kernel.

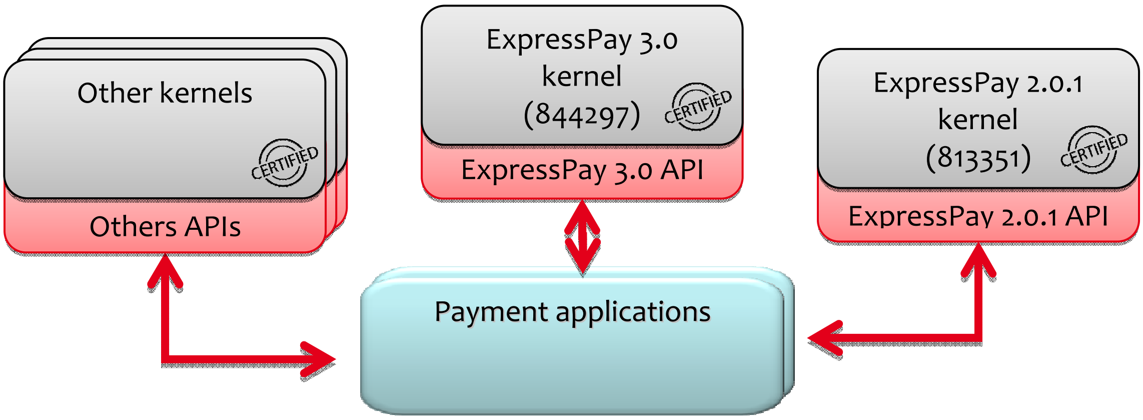
The changes could be summarised as follow:

- New standalone kernel.
- New tags.
- New value returned by the ExpressPay3_DoTransaction() function.

2. New standalone kernel

The ExpressPay 3 kernel is a new application.
It has a new application ID (844297) and a new application type (AD09).

The ExpressPay 3 kernel can cohabitate with ExpressPay 2 within the same terminal.



If you are using the new ExpressPay 3 kernel and if the ExpressPay 2 kernel is not needed by your application, you can remove it from the terminal.

APIs have been changed to *ExpressPay3_xxxx()* functions.
For example, an application must call the *ExpressPay3_DoTransaction()* function to perform an ExpressPay 3 transaction.

There is no backward compatibility between ExpressPay 2.0.1 and ExpressPay 3.0.

3. New tags

Some new tags must be sent to the kernel when calling the *ExpressPay3_DoTransaction()* function.

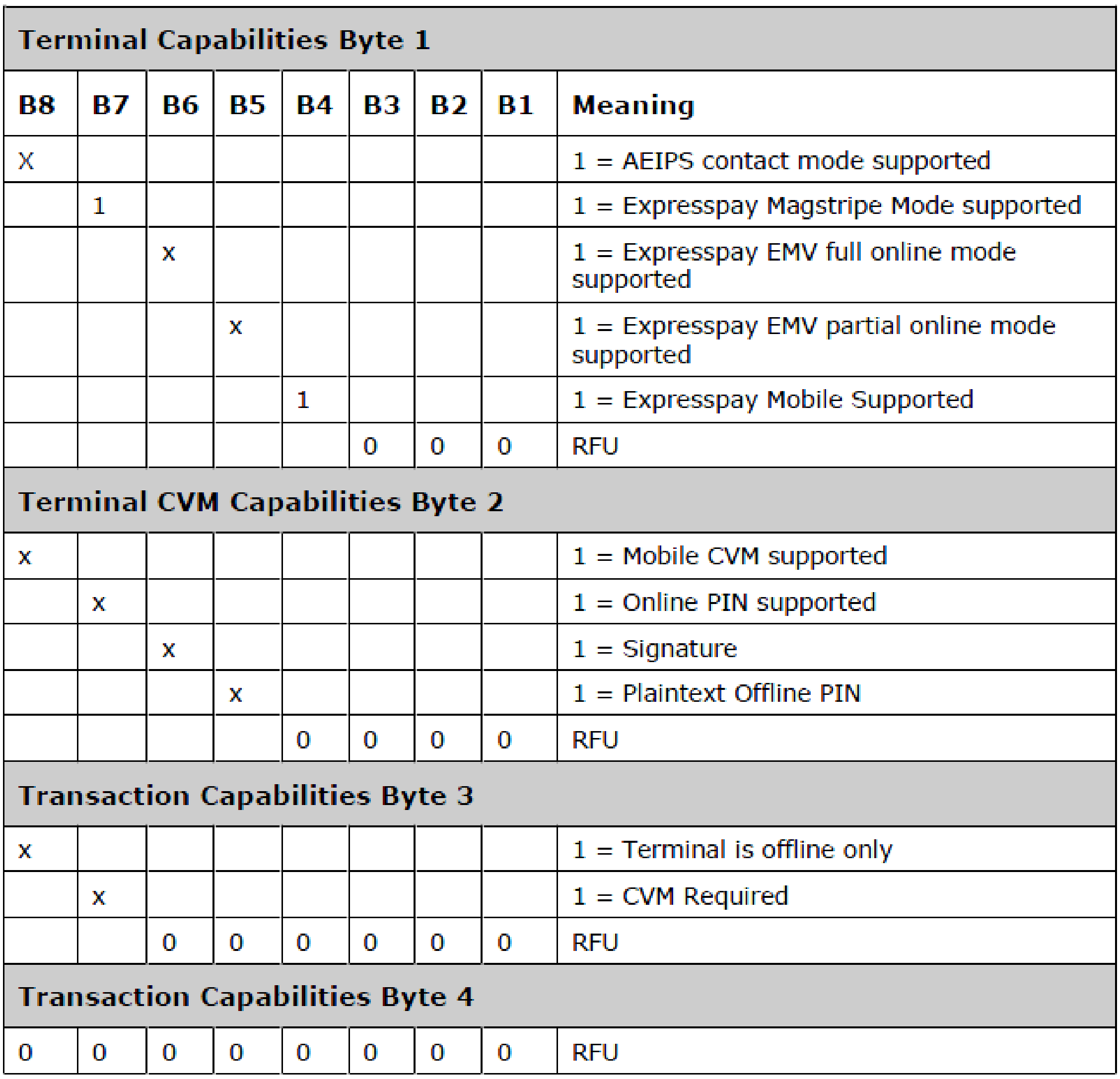
3.1. Terminal Transaction Capabilities

TAG_EXPRESSPAY_TERMINAL_TRANSACTION_CAPABILITIES (0x9F6E)

This data is used for managing ExpressPay transactions, and includes ExpressPay terminal capabilities (static data): Magstripe / EMV mode / Mobile / Contact / CVMs… and transaction dynamic data: CVM required…

The Byte 3 must be set to 0x00 and will be modified by the ExpressPay kernel during the transaction.
All other Terminal Transaction Capabilities settings (bytes 1, 2 and 4) are defined at Terminal configuration.

Terminal Transaction Capabilities - EMV Tag '9F6E'



Terminal Capabilities Byte 1								
B8	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**Meaning**
X								1 = AEIPS contact mode supported
	1							1 = Expresspay Magstripe Mode supported
		x						1 = Expresspay EMV full online mode supported
			x					1 = Expresspay EMV partial online mode supported
				1				1 = Expresspay Mobile Supported
					0	0	0	RFU
Terminal CVM Capabilities Byte 2								
x								1 = Mobile CVM supported
	x							1 = Online PIN supported
		x						1 = Signature
			x					1 = Plaintext Offline PIN
				0	0	0	0	RFU
Transaction Capabilities Byte 3								
x								1 = Terminal is offline only
	x							1 = CVM Required
		0	0	0	0	0	0	RFU
Transaction Capabilities Byte 4								
0	0	0	0	0	0	0	0	RFU

Note that the support of EMV Full Online mode is indicated in the Byte 1 Bit 6 and that there is no need to send the TAG_EXPRESSPAY_FULL_ONLINE_EMV_REMOVAL_TIMEOUT tag to the ExpressPay kernel anymore. This tag is obsolete. A removal timeout value is just used by your payment application to manage the online removal timeout.

Note that the support of contact interface is indicated in the Byte 1 Bit 8 and that there is no need to send the TAG_EXPRESSPAY_CONTACT_EMV_CAPABLE tag to the ExpressPay kernel anymore. This tag is obsolete.

If these two tags are sent to the kernel, they will not be used.

3.2. Mobile transaction restarted indicator

TAG_EXPRESSPAY_MOBILE_TRANSACTION_RESTARTED (0x9F918213)

When the ExpressPay kernel returns KERNEL_STATUS_MOBILE to the *ExpressPay3_DoTransaction()* function, Mobile CVM must be performed and then the transaction is restarted with another call to *ExpressPay3_DoTransaction()*. In this case, the payment application must send the TAG_EXPRESSPAY_MOBILE_TRANSACTION_RESTARTED tag to the kernel to indicate that the transaction has been restarted (value must be different from 0x00).

Please read the ExpressPay kernel documentation from the *Easy Path to C'Less* package for more details about how to perform Mobile CVM.

3.3. List of CA Public Key Index supported

TAG_EXPRESSPAY_INT_SUPPORTED_CAPK_INDEX_LIST (0x9F918216)

This tag contains the list of all CA Public Key index supported by your payment application for the selected AID. It is the concatenation of indexes (each one coded on one byte). This tag can contain up to 64 supported key indexes.

It must be sent to the ExpressPay kernel when calling the *ExpressPay3_DoTransaction()* function.

3.4. Terminal Type modified

When your application performed a transaction with the ExpressPay 2 kernel, the Terminal Type sent to the kernel was a modified Terminal Type: 0x9F35 tag modified with the content of the ExpressPay Terminal Capabilities (0x9F6D tag).

When performing an ExpressPay 3 transaction, your payment application does not need to send the modified Terminal Type to the ExpressPay kernel anymore. The application must send the original Terminal Type (0x9F35 tag) and the ExpressPay Terminal Capabilities (0x9F6D tag) to the kernel. The kernel will automatically modify the Terminal Type when requested in a DOL by the Card.

4. New values returned by ExpressPay3_DoTransaction() function

4.1. Mobile CVM processing

The specification defines the “Try Again” processing that will allow the cardholder to satisfy the Mobile CVM requirement out-of-field and to restart a transaction.

The Try Again outcome is required if Cardholder Verification processing determines that Mobile CVM is required and has not been successfully performed as indicated by the Mobile CVM Results in the GPO response or Status Word ‘6984’ in the GENERATE AC response. In these cases, the ExpressPay kernel will return KERNEL_STATUS_MOBILE to the ExpressPay3_DoTransaction() function.

Then the application will allow the cardholder to satisfy the Mobile CVM requirement out of field and restart the transaction:

- The cardholder is notified to withdraw and/or consult the mobile device (Card).
- The contactless reader field must be deactivated for a period of time (configurable in the range of one to three seconds, default value to be 1.5 seconds).
- The Mobile CVM is performed (code entered by the user via the phone).
- The transaction is restarted from Protocol Activation (detection/selection). The context of the current transaction persists unchanged.
- The cardholder is notified to re-present the mobile device.
- An indicator is set to identify that a transaction has been restarted: TAG_EXPRESSPAY_MOBILE_TRANSACTION_RESTARTED tag is sent to the kernel.

Please read the ExpressPay kernel documentation from the *Easy Path to C’Less* package for more details about how to perform Mobile CVM.