EMVL2 Application Library

Reference Manual

Version 1.24

July 2015

http://www.castech.com.tw

CASTLES TECHNOLOGY

Doc.#

Confidential Level: High

Table of Contents

Revision History

Version	Date	Description
V1.0	2013.8.30	Release
V1.1	2013.10.16	1. Add EMV_SpecialEventRegister function 2. Add EVENT_OUPUTCARDAPDU.
V1.2	2014.03.25	1. Add EMV_TxnDataSet function 2. Add EMV_ApplicationListSet function
V1.21	2014.03.31	Wording Amended.
V1.22	2014.04.07	Wording Amended.
V1.23	2014.06.17	Wording Amended.
V1.24	2015.07.03	Wording Amended.

WARNING

Information in this document is subject to change without prior notice.
No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Castles Technology Co., Ltd.
All trademarks mentioned are proprietary of their respective owners.

ABOUT THIS MANUAL

1 Introduction

EMVL2 Application Library is based on EMV Level 2 Library to provide a more-friendly, structured, and easy-use interface for EMV application development. In addition, this library uses XML format as its configuration/setting file to take advantage for EMV data maintenance (reading/updating).

2 Configuration

This library shall be used along with configuration file. The configuration file is for storing the terminal/application data, application identifier list and CAPK(Certification Authority Public Key). For the format and content of configuration file, please refer to the document “EMVL2 Application Configuration Reference Manual”.

Note that the configuration file shall exist within the user application before using this library. It can be loaded into the terminal along with the user application.

3 Structures

3.1 EMV_EVENT

This structure consists of a set of callback functions used by EMVL2 Application library.

STRUCTURE:		
EMV_EVENT		
MEMBER		
Version	Format version, should be 0x01.	
OnDisplayShow	O	
OnErrorMsg	O	
OnEMVConfigActive	O	
OnHashVerify	O	
OnTxnDataGet	M	
OnAppList	M	
OnAppSelectedConfirm	M	
OnTerminalDataGet	O	
OnCAPKGet	O	Callback functions. Detailed descriptions are
OnGetPINNotify	M	illustrated in chapter 5 and chapter 6.
OnOnlinePINBlockGet	C	(M : Mandatory, O : Optional, C : Conditional)
OnOfflinePINBlockGet	C	
OnOfflinePINVerifyResult	O	
OnTxnOnline	M	
OnTxnIssuerScriptResult	O	
OnTxnResult	M	
OnTotalAmountGet	O	
OnExceptionFileCheck	O	
OnCAPKRevocationCheck	O	

3.2 EMV_TXNDATA

STRUCTURE:
EMV_TXNDATA
MEMBER

[IN] Version	Format version, should be 0x01.
[IN] ulAmount	Transaction amount
[IN] bPOSEntryMode	Point-of-Service Entry Mode. The default value is 0x00.
[IN] bTxnType	Transaction Type
[IN] TxnDate	Transaction Date, format is “YYMMDD”
[IN] TxnTime	Transaction Time, format is “HHMMSS”
[IN] isForceOnline	Let transaction to force online.

3.3 DEFAULT_GETPIN_FUNC_PARA

STRUCTURE:	
DEFAULT_GETPIN_FUNC_PARA	
MEMBER	
[IN] Version	Format version, should be 0x01
[IN] usLineLeft_X	Specified asterisk left X position.
[IN] usLineRight_X	Specified asterisk right X position.
[IN] usLinePosition_Y	Specified asterisk Y position.
[IN] ulTimeout	Waiting Time in seconds for PIN entry (0means infinite).
[IN] bPINDigitMaxLength	Maximum number of PIN digits.
[IN] bPINDigitMinLength	Minimum number of PIN digits.
[IN] IsRightToLeft	Specified asterisk is Right-Justified or Left-Justified. 0 is Left-Justified, 1 is Right –Justified.
[IN] IsReverseLine	Specified line during left X to right X is reverse.
[IN] ONLINEPIN_PARA.CipherKeySet	Key set value of a KMS2 key for encrypting PIN.
[IN] ONLINEPIN_PARA.CipherKeyIndex	Key index value of a KMS2 key for encrypting PIN.
[IN] ONLINEPIN_PARA.bPANLen	Personal Account Number string length from 8 to 19.
[IN] ONLINEPIN_PARA.baPAN	Pointer for the bytes containing Personal Account Number to be used. For example, if the PAN is "401234567890”, the LenPAN is equal to 12. Note that the check digit of PAN shall be included.

3.4 ONLINE_PIN_DATA

STRUCTURE:

ONLINE_PIN_DATA	
MEMBER	
[IN] isOnlinePINRequired	Online PIN required flag.
[IN] pPIN	Online PIN Block
[IN] bPINLen	Length of online PIN block.

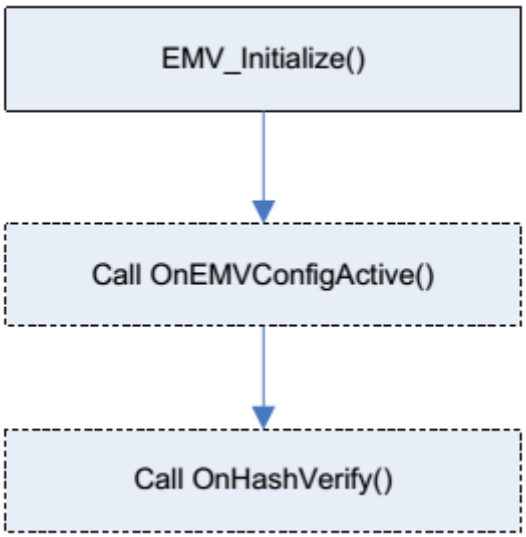
3.5 EMV_ONLINE_RESPONSE_DATA

STRUCTURE:	
EMV_ONLINE_RESPONSE_DATA	
MEMBER	
[OUT] bAction	Online result Refers to Definition d_ONLINE_ACTION_APPROVAL (0x01) d_ONLINE_ACTION_DECLINE (0x02) d_ONLINE_ACTION_UNABLE (0x03) d_ONLINE_ACTION_ISSUER_REFERRAL_APPR (0x04) d_ONLINE_ACTION_ISSUER_REFERRAL_DENY(0x05)
[OUT] pAuthorizationCode	Response code from the host.
[OUT] pIssuerAuthenticationData	Issuer Authentication Data from the issuer host.
[OUT] IssuerAuthenticationDataLen	Length of the Issuer Authentication Data.
[OUT] pIssuerScript	Issuer Script from the issuer host.
[OUT] IssuerScriptLen	Length of the Issuer Script.

3.6 EMV_APPLICATION_PARA

STRUCTURE:	
EMV_APPLICATION_PARA	
MEMBER	
pbAID	Application Identifier
bAIDLen	Length of the Application Identifier
bApplication_Selection_Indicator	Application Selection Indicator

4 Application Initialize



4.1 EMV_Initialize

Perform the initialization of EMVL2 Application library. The function will do the setting for the event functions as well as loading the configuration file.

C SYNTAX:		
USHORT EMV_Initialize(IN EMV_EVENT *pEvent, IN char* pEMVConfigFileName);		
PARAMETERS:		
[IN] pEvent	Point to an EMV_EVENT structure.	
[IN] pEMVConfigFileName	Configuration file name.	
RETURN VALUE:		
d_EMVAPLIB_OK	Success	
d_EMVAPLIB_ERR_DLL_LOAD_ERROR	Not have to load “libcaemvl2.so” or “libxml2.so”	
d_EMVAPLIB_ERR_EVENT_VERSION	Version on event structure is different, correct version should be 0x01.	
d_EMVAPLIB_ERR_EVENT_GET_TXNDATA	Not set OnTxnDataGet event	
d_EMVAPLIB_ERR_EVENT_ONLINE	Not set OnTxnOnline event	
d_EMVAPLIB_ERR_EVENT_SELECTED	Not set OnAppList event	
d_EMVAPLIB_ERR_EVENT_CONFIRMED	Not set OnAppSelectedConfirm event	
d_EMVAPLIB_ERR_EVENT_TXN_RESULT	Not set OnTxnResult event	
d_EMVAPLIB_ERR_EVENT_GETPIN_NOTIFY	Not set OnGetPINNotify event	
d_EMVAPLIB_ERR_CONFIG_READ_FAIL	Read configure file failure	
d_EMVAPLIB_ERR_NO_ACTIVE_INDEX	No active index of the configure	

d_EMVAPLIB_ERR_SET_TERMINAL_DATA_ERROR	Set terminal data failure
d_EMVAPLIB_ERR_SET_APPLIST_ERROR	Set application list failure

REMARKS:

EXAMPLE:

4.2 OnEMVConfigActive

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used to set the active configuration. If not providing this function, the active configuration is according to the configuration file.

C SYNTAX:	
void (*OnEMVConfigActive)(INOUT BYTE* pActiveIndex);	
PARAMETERS:	
[INOUT] pActiveIndex	[IN] Current Active Index. [OUT] The index of Config to be active
RETURN VALUE:	
	None
REMARKS:	
EXAMPLE:	

4.3 OnHashVerify

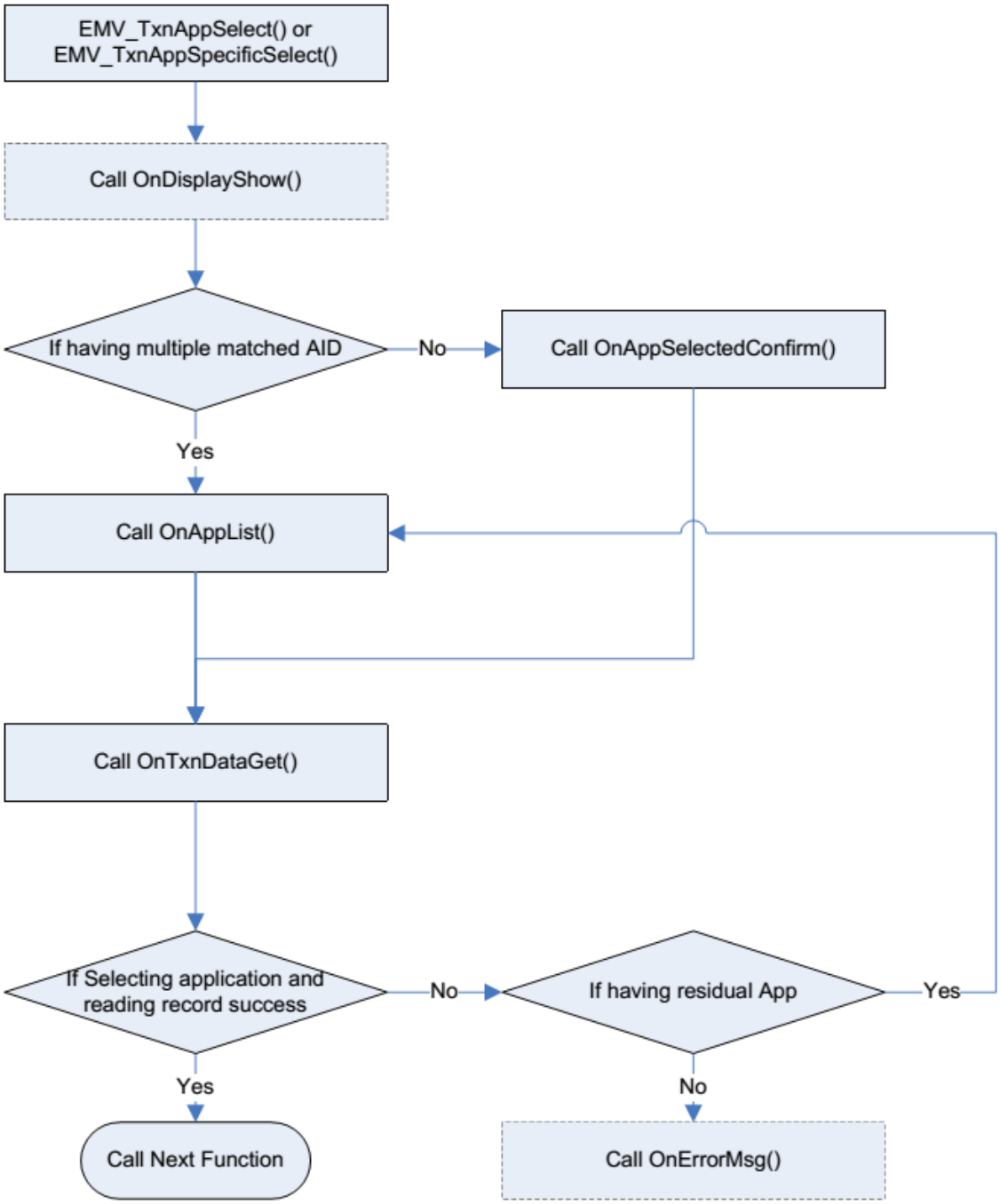
Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used for application to verify if the CAPKs of EMV kernel are correct.

C SYNTAX:	
BOOL (*OnHashVerify)(IN BYTE *pRID, IN BYTE F, IN BYTE *pModulus, IN USHORT ModulusLen, IN BYTE *pExponent, IN USHORT ExponentLen, IN BYTE *pHash, IN USHORT HashLen);	
PARAMETERS:	
[IN] pRID	Registered Application Provider Identifier
[IN] pModulus	Certification Authority Public Key Index
[IN] ModulusLen	Certification Authority Public Key Modulus
[IN] pExponent	Length of the Certification Authority Public Key Modulus
[IN] ExponentLen	Certification Authority Public Key Exponent
[IN] pHash	Hash
[IN] HashLen	Length of the Hash
RETURN VALUE:	
TRUE	The CAPK is correct.
FALSE	The CAPK is incorrect.
REMARKS:	
EXAMPLE:	

5 Transaction Start

After EMV kernel initialized, the transaction can be performed.



5.1 EMV_TxnAppSelect

The first step of performing transaction is Select Application after card inserted. If the card returns to indicate it has multiple applications, the event function OnAppList( ) will be triggered to provide those Application Labels for the application. The application then is required to show on the screen for user selecting. If the card has only one application, the event function OnAppSelectedConfirm will be triggered with indicating if the card requires the user to do the confirmation.

C SYNTAX:

USHORT EMV_TxnAppSelect(OUT BYTE* pSelectedAID, OUT BYTE* pSelectedAIDLen, OUT BYTE *pSelectedAppLabel, OUT BYTE *pSelectedAppLabelLen);

PARAMETERS:

[OUT] pSelectedAID	The Selected Application Identifier.
[OUT] pSelectedAIDLen	The length of the Application Identifier.
[OUT] pSelectedAppLabel	The Selected Application Label.
[OUT] pSelectedAppLabelLen	The length of the Application Label.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_SET_DEFAULT_DATA_ERROR	Not set terminal data to terminal buffer.
d_EMVAPLIB_ERR_TERM_DATA_MISSING	Terminal data miss
d_EMVAPLIB_ERR_ONLY_1_AP_NO_FALLBACK	Only one application and its blocked
d_EMVAPLIB_ERR_FUNCTION_NOT_SUPPORTED	Function not supported
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error
d_EMVAPLIB_ERR_DATA_BUFFER_EXCEEDED	Buffer size exceed
d_EMVAPLIB_ERR_NO_AP_FOUND	Application not found
d_EMVAPLIB_ERR_EVENT_CONFIRMED	Not set OnAppSelectedConfirm event
d_EMVAPLIB_ERR_EVENT_SELECTED	Not set OnAppList event
d_EMVAPLIB_ERR_EVENT_GET_TXNDATA	Not set OnTxnDataGet event
d_EMVAPLIB_ERR_EVENT_VERSION	Version on EMV_TXNDATA structure is different, correct version should be 0x01.
Other	Return directly the error code of the callback functions(OnAppSelectedConfirm, OnAppList and OnTxnDataGet).

REMARKS:

EXAMPLE:

5.2 EMV_TxnAppSpecificSelect

Similar with EMV_TxnAppSelect function, but Application Label is provided by the application. This function will do the match for the input Application Label with the Application Label returned by card. If not matched, the function will terminate with returning error code.

C SYNTAX:

USHORT EMV_TxnAppSpecificSelect(IN BYTE* pSpecificAppLabel, IN BYTE bSpecificAppLabelLen);

PARAMETERS:

[IN] pSpecificAppLabel	The Specific Application Label.
[IN] bSpecificAppLabelLen	The length of the Application Label.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_SET_DEFAULT_DATA_ERROR	Not set terminal data to terminal buffer.
d_EMVAPLIB_ERR_TERM_DATA_MISSING	Terminal data miss
d_EMVAPLIB_ERR_ONLY_1_AP_NO_FALLBACK	Only one application and its blocked
d_EMVAPLIB_ERR_FUNCTION_NOT_SUPPORTED	Function not supported
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error
d_EMVAPLIB_ERR_DATA_BUFFER_EXCEEDED	Buffer size exceed
d_EMVAPLIB_ERR_NO_AP_FOUND	Application not found
d_EMVAPLIB_ERR_EVENT_GET_TXNDATA	Not set OnTxnDataGet event
d_EMVAPLIB_ERR_EVENT_VERSION	Version on EMV_TXNDATA structure is different, correct version should be 0x01.
Other	Return directly the error code of the callback functions(OnTxnDataGet).

REMARKS:

EXAMPLE:

5.3 OnDisplayShow

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used for the application to show the message returned by the library during the transaction.

C SYNTAX:		
void (*OnDisplayShow)(IN char *pStrMsg);		
PARAMETERS:		
[IN] pStrMsg	Display message	
	d_EN_MSG_PROCESSING	"Processing..."
	d_EN_MSG_SELECT_FAIL	"Select Fail\nPlz Try Again"
RETURN VALUE:		
	None	
REMARKS:		
EXAMPLE:		

5.4 OnAppList

Mandatory. A event (callback) function provided by the user application for EMVL2 Application library using.

When the card has multiple applications, this event function will be triggered to provide Application Labels for the application to show on the screen to let user do the selection.

C SYNTAX:	
USHORT (*OnAppList)(IN BYTE AppNum, IN char AppLabel[ ][d_LABEL_STR_SIZE+1], OUT BYTE *pAppSelectedIndex);	
PARAMETERS:	
[IN] AppNum	Number of the application labels.
[IN] AppLabel	Array of the application labels. Every array size is 17bytes.
[OUT] pAppSelectedIndex	Output selected application index, range of value is 0 to AppNum-1.
RETURN VALUE:	
d_EMVAPLIB_OK	Success
Others	Defined by application. The value other than d_EMVAPLIB_OK will make EMV_TxnAppSelect function terminate and return this value.
REMARKS:	
EXAMPLE:	

5.5 OnAppSelectedConfirm

Mandatory. A event (callback) function provided by the user application for EMVL2 Application library using.

When the card has only one application, this event function will be triggered with card requiring confirmation indicator for the application to show on the screen to let user do the confirmation.

C SYNTAX:	
USHORT (*OnAppSelectedConfirm)(IN BOOL IsRequiredbyCard, IN BYTE *pLabel, IN BYTE bLabelLen);	
PARAMETERS:	
[IN] IsRequiredbyCard	Indicate if the confirmation action is required by card
[IN] pLabel	Confirmed application label.
[IN] bLabelLen	Length of the application label.
RETURN VALUE:	
d_EMVAPLIB_OK	Success
Definition	Defined by application. The value other than d_EMVAPLIB_OK will make EMV_TxnAppSelect function terminate and return this value.
REMARKS:	
EXAMPLE:	

5.6 OnTxnDataGet

Mandatory. A event (callback) function provided by the user application for EMVL2 Application library using.

After application selection, this event function will be triggered for the application to provide the transaction data, such as transaction amount, date, time, etc.

C SYNTAX:	
USHORT (*OnTxnDataGet)(OUT EMV_TXNDATA *pTxnData);	
PARAMETERS:	
[OUT] pTxnData	Data structure for transaction, description refers to below table.
RETURN VALUE:	
d_EMVAPLIB_OK	Success
Definition	Defined by application. The value other than d_EMVAPLIB_OK will make EMV_TxnAppSelect function terminate and return this value.
REMARKS:	
EXAMPLE:	

STRUCTURE:	
EMV_TXNDATA	
MEMBER	
Version	Structure version, current version is 0x01.
ulAmount	Transaction amount
bPOSEntryMode	Point-of-Service Entry Mode, default value is 0x00
bTxnType	Transaction Type
TxnDate	Transaction Date, format is “YYMMDD”
TxnTime	Transaction Time, format is “HHMMSS”
isForceOnline	Let transacton to force online.

5.7 OnErrorMsg

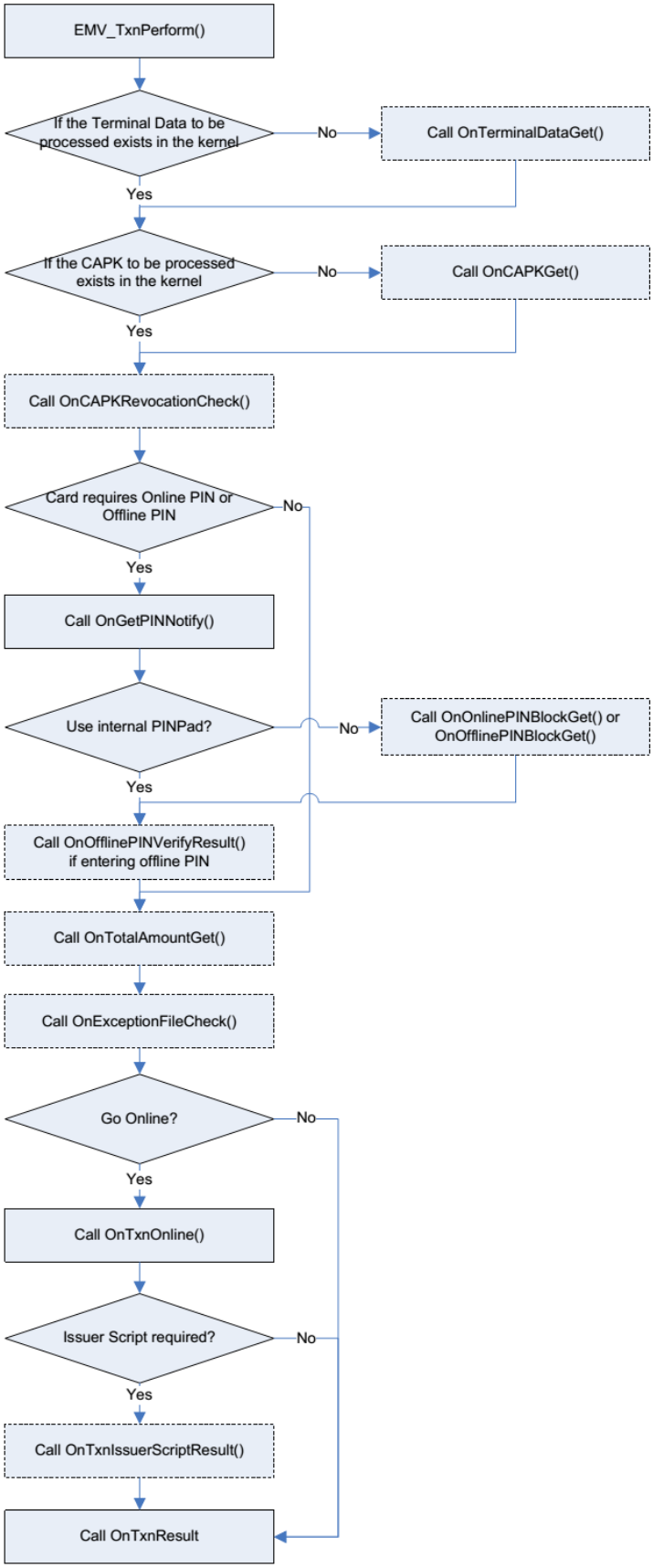
Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

If an error occurs during Select Application, this event function will be triggered for the application to show the error string message.

C SYNTAX:		
void (*OnErrorMsg)(IN char *pStrMsg);		
PARAMETERS:		
[IN] pStrMsg	Error message.	
	d_EN_MSG_PROCESSING	"Processing..."
	d_EN_MSG_SELECT_FAIL	"Select Fail\nPlz Try Again"
RETURN VALUE:		
	None	
REMARKS:		
EXAMPLE:		

6 Transaction Perform

The transaction will perform “Card Authentication” and link back to the host after “Select Application” and “Read Card Data” are finished.



6.1 EMV_TxnPerform

This function shall be called to perform transaction after EMV_TxnAppSelect or EMV_TxnAppSpecificSelect finished. This function will perform the actions based EMV specification, such as Initiate Application Processing, Read Application Data, Offline Authentication, Processing restrictions, Cardholder Verification, Terminal Risk Management, Terminal Action Analysis, Online Processing, Issuer Script Processing, etc. During the period of this function processing, required event functions will be triggered for the application to provide the necessary data or do the required actions, such as the retrieving for PIN block, CAPK, Terminal Data, or blacklist, or connecting with the host, etc.

C SYNTAX:	
USHORT EMV_TxnPerform(void);	
PARAMETERS:	
None	
RETURN VALUE:	
d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error
d_EMVAPLIB_ERR_GAC1_6985_FALLBACK	GenAC1 return status 0x6985
d_EMVAPLIB_ERR_FORCE_ACCEPTANCE	Force acceptance
d_EMVAPLIB_ERR_EVENT_ONLINE	Not set OnTxnOnline event
d_EMVAPLIB_ERR_EVENT_BUFFER_DEFINE	Not set pAuthorizationCode, pIssuerAuthenticationData and pIssuerScript buffer size.
d_EMVAPLIB_ERR_EVENT_TXN_RESULT	Not set OnTxnResult event
d_EMVAPLIB_ERR_SEND_APDU_CMD_FAIL	Send APDU command is failure.
d_EMVAPLIB_ERR_ERROR_9F4A_RULE	Tag 9F4A rule error
d_EMVAPLIB_ERR_KEY_NO_FOUND	Key not found
d_EMVAPLIB_ERR_PAN_NOT_SAME	The PAN isn't the same.
d_EMVAPLIB_ERR_DDOL_MISS	The DDOL is lost.
d_EMVAPLIB_ERR_INTERNAL_AUTHENTICATE_FAIL	Internal authenticate fail.
d_EMVAPLIB_ERR_NO_OFFLINE_DATA_AUTH_MATCH	Not match offline data authentication method
d_EMVAPLIB_ERR_MISS_APP_EXPIRATION_DATE	The application effective date is lost
d_EMVAPLIB_ERR_CARDHOLDER_VER_NOT_SUPP	Cardholder verification isn't supported.
d_EMVAPLIB_ERR_ERR_CVM_LIST_MISSING	CVM list is lost.
d_EMVAPLIB_ERR_GET_DATA_CMD_ERROR	Get data command error.
d_EMVAPLIB_ERR_DIS_TLV_TAG_ZERO	TLV tag second byte is zero.

d_EMVAPLIB_ERR_DIS_TLV_FAIL	TLV error when dismantling.
d_EMVAPLIB_ERR_DIS_TLV_EXCEED_MAX_LEN	Tag value length over the total length.
d_EMVAPLIB_ERR_DDOL_NOT_HAVE_9F37	DDOL have not tag 9F37.
d_EMVAPLIB_ERR_SDA_DATA_ERROR	Static Data Authentication data is error.
d_EMVAPLIB_ERR_SDA_ALGORITHM_NOT_SUPPORT	Static Data Authentication algorithm is not support.
d_EMVAPLIB_ERR_DDA_DATA_ERROR	Dynamic Data Authentication data is error.
d_EMVAPLIB_ERR_DDA_ALGORITHM_NOT_SUPPORT	Dynamic Data Authentication algorithm is not support.
d_EMVAPLIB_ERR_ISSUER_CERT_NOT_EXIST	Issuer Public Key Certificate not existed.
d_EMVAPLIB_ERR_ISSUER_CERT_FORMAT_ERROR	Issuer Public Key Certificate format is error.
d_EMVAPLIB_ERR_ISSUER_CERT_IIN_PAN_NOT_SAME	Issuer identification number and PAN not same.
d_EMVAPLIB_ERR_ISSUER_CERT_REVOCATION_FOUND	Issuer Public Key Certificate revocation found
d_EMVAPLIB_ERR_ISSUER_CERT_ALGORITHM_NOT_SUPPORT	Issuer Public Key Certificate algorithm not support.
d_EMVAPLIB_ERR_ISSUER_CERT_LENGTH_ERROR	Issuer Public Key Certificate length is error.
d_EMVAPLIB_ERR_ISSUER_CERT_EXPIRATION_DATE	Issuer Public Key Certificate has expired.
d_EMVAPLIB_ERR_ISSUER_CERT_HASH_NOT_MATCH	Issuer Public Key Certificate hash is not mach.
d_EMVAPLIB_ERR_ISSUER_CERT_EXPONENT_NOT_EXIST	Issuer Public Key Certificate exponent is not exist.
d_EMVAPLIB_ERR_ISSUER_CERT_REMAINDER_MISSING	Issuer Public Key Certificate pomander is lost.
d_EMVAPLIB_ERR_READ_DATA_TAG_NOT_70	Read tag data error, not 70 format
d_EMVAPLIB_ERR_ICC_CERT_NOT_EXIST	ICC Public Key Certificate not existed.
d_EMVAPLIB_ERR_ICC_CERT_FORMAT_ERROR	ICC Public Key Certificate format error
d_EMVAPLIB_ERR_ICC_CERT_ALGORITHM_NOT_SUPPORT	ICC Public Key Certificate algorithm is not support.
d_EMVAPLIB_ERR_ICC_CERT_LENGTH_ERROR	ICC Public Key Certificate length is error.
d_EMVAPLIB_ERR_ICC_CERT_HASH_NOT_MATCH	ICC Public Key Certificate hash is not mach.
d_EMVAPLIB_ERR_ICC_CERT_EXPIRATION_DATE	ICC Public Key Certificate has expired.
d_EMVAPLIB_ERR_ICC_CERT_EXPONENT_NOT_EXIST	ICC Public Key Certificate exponent is not existed.
d_EMVAPLIB_ERR_ICC_ISSUER_PK_NOT_EXIST	ICC Public Key Certificate PK is not existed.

d_EMVAPLIB_ERR_ICC_CERT_REMAINDER_MISSING	ICC Public Key Certificate pomander is lost.
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_PIN_NOT_KEYIN	Plaintext PIN verification performed by ICC but cardholder didn't input the PIN.
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_PIN_TRY_LIMIT_EXCEEDED	PIN Try Limit exceeded.
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_PIN_OK	Input plaintext PIN verification has successfully
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_PIN_WRONG	Input plaintext PIN verification has wrong.
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_UNKNOW_SW12	The responded data of APDU is unknown when input plaintext PIN verification.
d_EMVAPLIB_ERR_CVM_TERMINAL_NOT_SUPPORT_SPECIFY_CVM	The terminal isn't support specify CVM.
d_EMVAPLIB_ERR_CVM_ENC_PIN_ONLINE_PIN_NOT_KEYIN	Encipher PIN online verification performed by ICC but cardholder didn't input the PIN.
d_EMVAPLIB_ERR_CVM_ENCIPHERED_PIN_NOT_KEYIN	Encipher PIN verification performed by ICC but cardholder didn't input the PIN.
d_EMVAPLIB_ERR_CVM_ENCIPHERED_PIN_TRY_LIMIT_EXCEEDED	Encipher PIN verification performed by ICC but cardholder didn't input the PIN.
d_EMVAPLIB_ERR_CVM_ENCIPHERED_PIN_UNKNOW_SW12	The responded data of APDU is unknown when input Encipher PIN verification.
d_EMVAPLIB_ERR_CVM_ENCIPHERED_PIN_GET_RN_UNKNOW_SW12	The responded data of APDU is unknown when send "GET CHALLENGE" command by input Encipher PIN action.
d_EMVAPLIB_ERR_MISSING_TERMINAL_DATA	Missing terminal data
d_EMVAPLIB_ERR_CARD_DATA_MULTIPLE	Card data multiple
d_EMVAPLIB_ERR_ISSUER_CERT_CAPKI_NOT_EXIST	Issuer Public Key Certificate CAPKI is not exist.
d_EMVAPLIB_ERR_CVM_PLAIN_TEXT_PIN_BLOCK	The ICC has been blocked.

REMARKS:

EXAMPLE:

6.2 OnTerminalDataGet

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function will be triggered when certain terminal data are required by the EMV kernel but these data do not exist in the EMV kernel. The application has to provide the corresponding terminal data by this event function. It is suggested for application to provide this function.

C SYNTAX:	
BOOL (*OnTerminalDataGet)(IN USHORT usTag, INOUT USHORT *pLen, OUT BYTE *pValue);	
PARAMETERS:	
[IN] usTag	EMV Tag
[INOUT] pLen	[IN] Size of Value. [OUT] Actual Length of Value.
[OUT] pValue	Value for Tag.
RETURN VALUE:	
TRUE	Get terminal data successfully.
FALSE	Get terminal data failure.
REMARKS:	
EXAMPLE:	

6.3 OnCAPKGet

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function will be triggered when certain CAPK (Certification Authority Public Key) is required by the EMV kernel but it does not exist in the EMV kernel. The application has to provide the corresponding CAPK by this event function. It is suggested for application to provide this function.

C SYNTAX:	
BOOL (*OnCAPKGet)(IN BYTE *pRID, IN BYTE bKeyIndex, OUT BYTE *pModulus, OUT USHORT *pModulusLen, OUT BYTE *pExponent, OUT USHORT *pExponentLen);	
PARAMETERS:	
[IN] pRID	Registered Application Provider Identifier
[IN] bKeyIndex	Certification Authority Public Key Index
[OUT] pModulus	Certification Authority Public Key Modulus
[OUT] pModulusLen	Length of the Certification Authority Public Key Modulus
[OUT] pExponent	Certification Authority Public Key Exponent
[OUT] pExponentLen	Length of the Certification Authority Public Key Exponent
RETURN VALUE:	
TRUE	Get CAPK successfully.
FALSE	Get CAPK failure.
REMARKS:	
EXAMPLE:	

6.4 OnCAPKRevocationCheck

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

This function is used for application to indicate if the CAPK is in revocation.

C SYNTAX:	
BOOL (*OnCAPKRevocationCheck)(IN BYTE *pbRID, IN BYTE bCAPKIdx, IN BYTE *pbSerialNumuber);	
PARAMETERS:	
[IN] pRID	Registered Application Provider Identifier
[IN] bKeyIndex	Certification Authority Public Key Index
[IN] pbSerialNumuber	Certificate Serial Number
RETURN VALUE:	
TRUE	Indicate the specific CAPK is invalid.
FALSE	Indicate the specific CAPK is valid.
REMARKS:	
EXAMPLE:	

6.5 OnGetPINNotify

Mandatory. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function will be triggered when the card requires the user to enter the online or offline PIN. This function should return to indicate which type of pinpad.

C SYNTAX:

void (*OnGetPINNotify)(IN BYTE bPINType, IN USHORT bRemainingCounter, OUT BOOL* pIsInternalPINPad, OUT DEFAULT_GETPIN_FUNC_PARA *pGetPINFuncPara);

PARAMETERS:

[IN] bPINType	Online or Offline PIN #define d_NOTIFY_ONLINE_PIN 0 #define d_NOTIFY_OFFLINE_PIN 1
[IN] bRemainingCounter	Remaining PIN try counter if PIN type is offline PIN.
[OUT] IsInternalPINPad	Indicate the EMV Kernel if it used internal PINPad for user input PIN. If yes, EMV kernel will use its internal function for getting online/offline PIN with the pGetPINFuncPara. If no, the function OnOnlinePINBlockGet or OnOfflinePINBlockGet provided by user application will be called when the EMV kernel requests to get PIN.
[OUT] pPara	If set default PIN getting function already, the parameters would fill in the value. Default GetPIN Parameter structure, description refers to below table.

RETURN VALUE:

None

REMARKS:

EXAMPLE:

STRUCTURE:

DEFAULT_GETPIN_FUNC_PARA

MEMBER

Version	Structure version
usLineLeft_X	Specified asterisk left X position.

usLineRight_X	Specified asterisk right X position.
usLinePosition_Y	Specified asterisk Y position.
ulTimeout	Waiting Time in seconds for PIN entry (0means infinite).
bPINDigitMaxLength	Maximum number of PIN digits.
bPINDigitMinLength	Minimum number of PIN digits.
IsRightToLeft	Specified asterisk is Right-Justified or Left-Justified. 0 is Left-Justified, 1 is Right –Justified.
IsReverseLine	Specified line during left X to right X is reverse.
ONLINEPIN_PARA	Structure
CipherKeySet	Key set value of a KMS2 key for encrypting PIN.
CipherKeyIndex	Key index value of a KMS2 key for encrypting PIN.
bPANLen	Personal Account Number string length from 8 to 19.
baPAN	Pointer for the bytes containing Personal Account Number to be used. For example, if the PAN is "401234567890”, the LenP AN is equal to 12. Note that the check digit of PAN shall be included.

6.6 OnOnlinePINBlockGet

Conditional. A event (callback) function provided by the user application for EMVL2 Application library using.

This function is required if the external pinpad is used and the terminal has to support online PIN, or if the application would like to implement online pin generation in its own way instead of using internal function provided by EMV Kernel. It is used to provide online PIN block.

This function will be triggered only when the card requires online PIN and the argument IsInternalPINPad returned by OnGetPINNotify is set to FALSE.

C SYNTAX:

```
USHORT (*OnOnlinePINBlockGet)(OUT ONLINE_PIN_DATA *pOnlinePINData);
```

PARAMETERS:

[OUT] pOnlinePINData	Online PIN Data structure, description refers to below table.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_PIN_BY_PASS	PIN by pass
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error

REMARKS:

EXAMPLE:

STRUCTURE:

ONLINE_PIN_DATA

MEMBER

isOnlinePINRequired	Online PIN required flag.
pPIN	Online PIN Block
bPINLen	Length of online PIN block.

6.7 OnOfflinePINBlockGet

Conditional. A event (callback) function provided by the user application for EMVL2 Application library using.

This function is required only if the external pin is used and the terminal is required to support offline PIN.

During cardholder verification, after the user enters the PIN on the external pin and the external pin returns the encrypted pin block, the application should put this encrypted pin block into EMV kernel by the CTOS_KMS2DataEncrypt function with KMS2_DATAENCRYPTCIPHERMETHOD_EXTPIN_ECB cipher mode. The EMV kernel will perform the offline pin verification with the card.

C SYNTAX:

```
USHORT (*OnOfflinePINBlockGet)(void);
```

PARAMETERS:

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_PIN_BY_PASS	PIN by pass
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error
d_EMVAPLIB_ERR_ENTER_KMS_OFFLINEPIN	When Using CTOS_KMSbGetEncOfflinePIN() and CTOS_KMS2PINGet(), please use this return code。

REMARKS:

EXAMPLE:

6.8 OnOfflinePINVerifyResult

Optional. A event (callback) function provided by the user application for EMVL2 Application library using.

The function is used for the application to get the results of Offline PIN verification from EMV kernel.

C SYNTAX:		
void (*OnOfflinePINVerifyResult)(IN USHORT usResult);		
PARAMETERS:		
[IN] usResult	Result for entering Offline PIN.	
	Refers to Definition	
	#define d_PIN_RESULT_OK	1
	#define d_PIN_RESULT_FAIL	2
	#define d_PIN_RESULT_BLOCKED	3
	#define d_PIN_RESULT_FAILBLOCKED	4
RETURN VALUE:		
	None	
REMARKS:		
EXAMPLE:		

6.9 OnTotalAmountGet

Optional. An event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used for EMV kernel to get the total transaction amount of the PAN in the same batch for the checking of Floor Limits.

C SYNTAX:	
void (*OnTotalAmountGet)(IN BYTE *pPAN, IN BYTE bPANLen, OUT ULONG *pAmount);	
PARAMETERS:	
[IN] pPAN	Application Primary Account Number.
[IN] bPANLen	Length of the Application Primary Account Number.
[OUT] pAmount	Total transaction amount.
RETURN VALUE:	
	None
REMARKS:	
EXAMPLE:	

6.10 OnExceptionFileCheck

Optional. An event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used for EMV kernel to check if the card is in blacklist.

C SYNTAX:	
void (*OnExceptionFileCheck)(IN BYTE *pPAN, IN BYTE bPANLen, OUT BOOL *isException);	
PARAMETERS:	
[IN] pPAN	Application Primary Account Number.
[IN] bPANLen	Length of the Application Primary Account Number.
[OUT] isException	Exception flag
RETURN VALUE:	
	None
REMARKS:	
EXAMPLE:	

6.11 OnTxnOnline

Optional. An event (callback) function provided by the user application for EMVL2 Application library using.

This event function will be triggered when the transaction is going online. The application shall connect to the host, sending the transaction, and then receiving the response data.

For the offline transaction, the OnTxnResult event will be triggered without triggering this function.

When this function is not present, users must implement and finish online process by themselves, and then call the EMV_TxnCompletion function to complete transaction.

C SYNTAX:	
void (*OnTxnOnline)(IN ONLINE_PIN_DATA *pOnlinePINData, OUT EMV_ONLINE_RESPONSE_DATA* pOnlineResponseData);	
PARAMETERS:	
[IN] pOnlinePINData	Online PIN data structure, description refers to section 5.7 OnOnlinePINBlockGet.
[OUT] pOnlineResponseData	Online response data structure, description refers to below table.
RETURN VALUE:	
	None
REMARKS:	
EXAMPLE:	
STRUCTURE:	
EMV_ONLINE_RESPONSE_DATA	
MEMBER	
bAction	Online result Refers to Definition #define d_ONLINE_ACTION_APPROVAL 1 #define d_ONLINE_ACTION_DECLINE 2 #define d_ONLINE_ACTION_UNABLE 3 #define

	d_ONLINE_ACTION_ISSUER_REFERRAL_APPR 4
	#define
	d_ONLINE_ACTION_ISSUER_REFERRAL_DENY 5

pAuthorizationCode	Response code from the host.
pIssuerAuthenticationData	Issuer Authentication Data from the issuer host.
IssuerAuthenticationDataLen	Length of the Issuer Authentication Data.
pIssuerScript	Issuer Script from the issuer host.
IssuerScriptLen	Length of the Issuer Script.

6.12 OnTxnIssuerScriptResult

Optional. An event (callback) function provided by the user application for EMVL2 Application library using.

In case of the host responding to terminal with the Issuer Authentication Data or Issuer Script, the application should pass these data to EMV kernel in the OnTxnOnline function. After EMV kernel processing, this event function will be triggered for the application to have the result of issuer scripts.

C SYNTAX:

```
void (*OnTxnIssuerScriptResult)(IN BYTE* pScriptResult, IN USHORT pScriptResultLen);
```

PARAMETERS:

[IN] pScriptResult	Issuer Script Result.
[IN] pScriptResultLen	Length of the Issuer Script Result.

RETURN VALUE:

None

REMARKS:

EXAMPLE:

6.13 OnTxnResult

Mandatory. A event (callback) function provided by the user application for EMVL2 Application library using.

This event function is used to report the application the transaction result.

C SYNTAX:		
void (*OnTxnResult)(IN BYTE bResult, IN BOOL IsSignatureRequired);		
PARAMETERS:		
[IN] bResult	Transaction Result.	
	Refers to Definition	
	#define d_TXN_RESULT_APPROVAL	0x01
	#define d_TXN_RESULT_DECLINE	0x02
[IN] IsSignatureRequired	Signature required flag.	
RETURN VALUE:		
	None	
REMARKS:		
EXAMPLE:		

6.14 EMV_TxnCompletion

If users do not implement the event function OnTxnOnline, instead, implement and finish online process by themselves, it is required to call this function to complete transaction.

The transaction result will be outputted from the OnTxnResult event.

C SYNTAX:	
USHORT EMV_TxnCompletion(IN EMV_ONLINE_RESPONSE_DATA* pOnlineResponseData);	
PARAMETERS:	
[IN] pOnlineResponseData	Online response data structure.
RETURN VALUE:	
d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_CRITICAL_ERROR	Critical error
d_EMVAPLIB_ERR_EVENT_TXN_RESULT	OnTxnResult event is set to NULL
d_EMVAPLIB_ERR_EVENT_BUFFER_DEFINE	pAuthorizationCode, pIssuerAuthenticationData or pIssuerScript is set to NULL.
REMARKS:	
EXAMPLE:	

7 Utility

7.1 EMV_DataGet

Get specific tag data (terminal data or card data) in the current transaction from EMV kernel.

C SYNTAX:	
USHORT EMV_DataGet(IN USHORT usTag, INOUT USHORT *pLen, OUT BYTE *pValue);	
PARAMETERS:	
[IN] usTag	EMV Tag.
[INOUT] pLen	[IN] Size of buffer
	[OUT] Actual length of the value
[OUT] pValue	Terminal data or card data
RETURN VALUE:	
d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_NO_DATA	Not found terminal or card data.
REMARKS:	
EXAMPLE:	

7.2 EMV_TerminalDataSet

Each AID of the AppList in Configuration file has its own terminal data area. This function is used to set the tag data into the specific AID terminal data area.

Note that this function won't change any setting of Configuration File.

C SYNTAX:

USHORT EMV_TerminalDataSet(IN BYTE* pAID, IN BYTE bAIDLen, IN USHORT usTag, IN USHORT usLen, IN BYTE *pValue);

PARAMETERS:

[IN] pAID	Application Identification.
[IN] bAIDLen	Length of the Application Identification.
[IN] usTag	EMV Tag.
[IN] usLen	Length of the value.
[IN] pValue	Terminal data.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_DATA_BUF_OVERFLOW	Buffer overflow.
d_EMVAPLIB_ERR_TXN_PROCESSING	Operation disallowed during transaction.

REMARKS:

EXAMPLE:

7.3 EMV_TerminalDataGet

Each AID of the AppList in Configuration file has its own terminal data area. This function is used to get the tag data from the specific AID terminal data area.

C SYNTAX:

USHORT EMV_TerminalDataGet(IN BYTE* pAID, IN BYTE bAIDLen, IN USHORT usTag, INOUT USHORT *pLen, OUT BYTE *pValue);

PARAMETERS:

[IN] pAID	Application Identifier
[IN] bAIDLen	Length of the Application Identifier.
[IN] usTag	EMV Tag.
[INOUT] usLen	[IN] Size of buffer
	[OUT] Actual length of the value
[OUT] pValue	Terminal data.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_DATA_BUFFER_ERR	Buffer not enough.
OR	
d_EMVAPLIB_ERR_NO_DATA	The terminal or card data not found.
d_EMVAPLIB_ERR_TXN_PROCESSING	Operation disallowed during transaction.

REMARKS:

EXAMPLE:

7.4 EMV_TerminalDataDelete

Each AID of the AppList in Configuration file has its own terminal data area. This function is used to remove the tag data from the specific AID terminal data area.

Note that this function won't change any setting of Configuration File

C SYNTAX:	
USHORT EMV_TerminalDataDelete(IN BYTE* pAID, IN BYTE bAIDLen, IN USHORT usTag);	
PARAMETERS:	
[IN] pAID	Application Identification.
[IN] bAIDLen	Length of the Application Identification.
[IN] usTag	EMV Tag.
RETURN VALUE:	
d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_NO_DATA	The terminal or card data not found.
d_EMVAPLIB_ERR_TXN_PROCESSING	Operation disallowed during transaction.
REMARKS:	
EXAMPLE:	

7.5 EMV_SpecialEventRegister

Register special event (callback) function to EMVLv2 Application library, provided by the user application using.

C SYNTAX:	
void EMV_SpecialEventRegister(IN BYTE EventID, IN void *EventFunc)	
PARAMETERS:	
[IN] EventID	Identifier for registering special event – d_EVENTID_OUTPUT_CARDAPDU
[IN] EventFunc	Register event (callback) function.
RETURN VALUE:	
	None
REMARKS:	
Event ID “d_EVENTID_OUTPUT_CARDAPDU“ must correspond with EVENT_OUTPUTCARDAPDU event function.	
EXAMPLE:	

7.5.1 EVENT_OUTPUTCARDAPDU

This event function is used to provide certain APDU commands during card communication for application developer debugging or card analysis.

To enable this event, it has to be registered by EMV_SpecialEventRegister with the event ID d_EVENTID_OUTPUT_CARDAPDU.

Note that these APDU commands “Verify”, “External Authentication” and “Issuer Script” will not be exported within this event function.

C SYNTAX:	
typedef void (*EVENT_OUTPUTCARDAPDU)(IN BYTE *pTxAPDU, IN USHORT TxAPUDLen, IN BYTE *pRxAPDU, IN USHORT RxAPDULen);	
PARAMETERS:	
[IN] pTxAPDU	APDU command sent from terminal to card.
[IN] TxAPUDLen	Length of APDU command.
[IN] pRxAPDU	APDU response from card
[IN] RxAPDULen	Length of APDU response.
RETURN VALUE:	
	None
REMARKS:	

EXAMPLE:

7.6 EMV_TxnDataSet

This function is used to set the data regarding to transaction, e.g. transaction amount, during transaction.

C SYNTAX:

USHORT EMV_TxnDataSet(IN USHORT usTag, IN USHORT usLen, IN BYTE *pValue);

PARAMETERS:

[IN] usTag	EMV Tag.
[IN] usLen	Length of the value.
[IN] pValue	Terminal data.

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_TXN_NOT_PROCESING	Operation allowed only during transaction.

REMARKS:

EXAMPLE:

7.7 EMV_ApplicationListSet

This function is used to add applications into the application list of the library.

It is required to provide a callback function OnSetAppData, which will be called back at the time EMV_ApplicationListSet called. If the OnSetAppData function returns TRUE, it indicates the library that there are rest applications required to be added into the list. So, in this case, the OnSetAppData function will be called again until it returns FALSE to indicate finish.

C SYNTAX:

USHORT EMV_ApplicationListSet(BOOL (*OnSetAppData)(OUT EMV_APPLICATION_PARA *pAppData));

PARAMETERS:

[IN] OnSetAppData	Callback function

RETURN VALUE:

d_EMVAPLIB_OK	Success
d_EMVAPLIB_ERR_EVENT_SETAPPDATA	The OnSetAppData event does not exist.
D_EMVAPLIB_ERR_DATA_BUFFER_EXCEEDED	Buffer not enough
d_EMVAPLIB_ERR_INVALID_PARA	Invalid parameter

REMARKS:

EXAMPLE:

STRUCTURE:

EMV_APPLICATION_PARA

MEMBER

pbAID	Application Identifier
bAIDLen	Length of the Application Identifier
bApplication_Selection_Indicator	Application Selection Indicator

8 Definition

Parameter	Value	Description
EMV Tag		
d_TAG_AID	0x004F	Application Identifier
d_TAG_TRACK2_EQUIVALENT_DATA	0x0057	Track2 Equivalent Data
d_TAG_PAN	0x005A	Application Primary Account Number
d_TAG_AMOUNT_AUTHORIZED_B	0x0081	Amount, Authorized (Binary)
d_TAG_ARC	0x008A	Authorization Code
d_TAG_TVR	0x0095	Terminal Verification Results
d_TAG_TRAN_DATE	0x009A	Transaction Date
d_TAG_TX_TYPE	0x009C	Transaction Type
d_TAG_TERM_CURRENCY_CODE	0x5F2A	Transaction Currency Code
d_TAG_TERM_CURRENCY_EXP	0x5F36	Transaction Currency Exponent
d_TAG_AMOUNT_AUTHORIZED	0x9F02	Amount, Authorized (Numeric)
d_TAG_AMOUNT_OTHER	0x9F03	Amount, Other (Numeric)
d_TAG_APP_VERSION	0x9F09	Application Version Number
d_TAG_TERM_COUNTRY_CODE	0x9F1A	Terminal Country Code
d_TAG_FLOOR_LIMIT	0x9F1B	Floor Limit
d_TAG_IFD_SN	0x9F1E	Interface Device Serial Number
d_TAG_TRAN_TIME	0x9F21	Transaction Time
d_TAG_TERM_CAP	0x9F33	Terminal Capabilities
d_TAG_TERM_TYPE	0x9F35	Terminal Type
d_TAG_UNPREDICATE_NUMBER	0x9F37	Unpredictable Number
d_TAG_POS_ENTRY_MODE	0x9F39	Point-of-Service Entry Mode
d_TAG_ADD_TERM_CAP	0x9F40	Additional Terminal Capabilities
d_TAG_TERM_SEQ_COUNTER	0x9F41	Transaction Sequence Counter
d_TAG_ISSUER_SCRIPT_RESULT	0x9F5B	Issuer Script Result
d_TAG_DEFAULT_TDOL	0xDFC0	Default TDOL
d_TAG_DEFAULT_DDOL	0xDFC1	Default DDOL
d_TAG_TARGET_PERCENT	0xDFC2	Target Percent
d_TAG_MAX_TARGET_PERCENT	0xDFC3	Max. Target Percent
d_TAG_THRESHHOLD_VALUE	0xDFC4	Threshold Value
d_TAG_TAC_DENIAL	0xDFC6	Terminal Action Code – Denial
d_TAG_TAC_ONLINE	0xDFC7	Terminal Action Code – Online
d_TAG_TAC_DEFAULT	0xDFC8	Terminal Action Code – Default
Buffer Size		

d_LABEL_STR_SIZE	16	Application Label Array Size
Transaction Result		
d_TXN_RESULT_APPROVAL	0x01	Transaction Approval
d_TXN_RESULT_DECLINE	0x02	Transaction Decline
d_TXN_RESULT_GO_ONLINE	0x03	Return Online on First Gen AC
Online Result		
d_ONLINE_ACTION_APPROVAL	1	Online Approval
d_ONLINE_ACTION_DECLINE	2	Online Decline
d_ONLINE_ACTION_UNABLE	3	Unable Go Online
d_ONLINE_ACTION_ISSUER_REFERRAL_APPR	4	Issuer Referral Approval
d_ONLINE_ACTION_ISSUER_REFERRAL_DENY	5	Issuer Referral Decline
PIN Verify Result		
d_PIN_RESULT_OK	1	Verify Success
d_PIN_RESULT_FAIL	2	Verify Failure
d_PIN_RESULT_BLOCKED	3	PIN Blocked
d_PIN_RESULT_FAILBLOCKED	4	

