Vanstone Common L2 Library Programming Manual

Vanstone Electronic (Beijing) Co.,Ltd.

Version

Item NO.	Version	Descriptions	Modefer/Date	Comments
1	V2.10	Create	Bruce Huang /2018.05.03	Initial draft

Table of Content

1 Return Codes Definition

The common definition of all the return codes are defined in this section, but the developer should refer to the separate API definition for more information.

Code	Macro	Definition
0	EMV_OK	Success
-2	ERR_EMVRSP	Command response error
-3	ERR_APPBLOCK	Application already blocked
-4	ERR_NOAPP	No more candidate for selection
-5	ERR_USERCANCEL	User canceled
-6	ERR_TIMEOUT	Operation timeout
-7	ERR_EMVDATA	Error data returned by card
-8	ERR_NOTACCEPT	Transaction not accepted, should terminate
-9	ERR_EMVDENIAL	Transaction declined
-10	ERR_KEYEXP	CA public key already expired
-11	ERR_NOPINPAD	PINpad malfunction
-12	ERR_NOPIN	Cardholder bypassed the PIN entry process
-13	ERR_CAPKCHECKSUM	CA public key checksum error
-14	ERR_NOTFOUND	Data object not found
-15	ERR_NODATA	Data object does not have any value
-16	ERR_OVERFLOW	Memory overflow

-21	ERR_ICCCMD	IC card command exchange error
-22	ERR_ICCBLOCK	Card already blocked
-25	ERR_USECONTACT	Please use contact interface instead
-26	ERR_APPEXP	Application already expired
-27	ERR_BLACKLIST	Card is detected in exception file
-32	ERR_UNSUPPORTED	Function not supported
-101	ERR_FILE	Terminal file operation failed
-102	ERR_PARAM	Parameter error
-103	ERR_PINBLOCK	Cardholder PIN blocked
-104	ERR_DATA_EXIST	Duplicated data entry
-105	ERR_AGAIN	Please try again
-111	ERR_ICCINSERTED	IC card inserted
-112	ERR_SEEPHONE	Please refer to the consumer devices (such as ApplePay or similar NFC mobile devices). Normally it requires the cardholder to tap again
-113	ERR_CLNOTALLOWED	All the Contactless Applications are Not Allowed
-114	ERR_SELECTNEXT	Error occurs during the current application, one need to select the next one in the application candidate list
-115	ERR_NOAMOUNT	No amount

2 COMMON_TERMINAL_PARAM(class)

2.1 Class Overview

COMMON _ TERMINAL_PARAM is a class which include the shared param of terminal to be configured.

2.2 Public constructor

COMMON _ TERMINAL_PARAM ()

2.3 Public member

Field Name	Definition
Byte[] MerchName = new byte[128]	Merchant Name and Location, Value of tag 9F4E as defined in the specification.
byte[] MerchCateCode = new byte[2]	Merchant Category Code, Value of tag 9F15 as defined in the specification.
byte[] MerchId = new byte[15]	Merchant Identifier, Value of tag 9F16 as defined in the specification.
byte[] TermId = new byte[8]	Terminal Identification, Value of tag 9F1C as defined in the specification
byte TerminalType	Terminal Type, Value of tag 9F35 as defined in the specification.
byte TransCurrExp	Transaction Currency Exponent, Value of tag 5F36 as defined in the specification.
byte ReferCurrExp;	Transaction Ref. Currency Exponent, Value of

	tag 9F3D as defined in the specification.
byte[] ReferCurrCode = new byte[2];	Transaction Ref. Currency Code, Value of tag 9F3C as defined in the specification.
byte[] CountryCode = new byte[2];	Terminal Country Code, Value of tag 9F1A as defined in the specification.
byte[] TransCurrCode = new byte[2];	transaction Currency Code, Value of tag 5F2A as defined in the specification.
long ReferCurrCon;	Ref. Currency
byte[] AcquierId = new byte[12];	Acquirer Identifier, Value of tag 9F01 as defined in the specification.
byte TransType	Transaction Type, Value of tag 9C as defined in the specification.

2 COMMON_PPSE_STATUS(class)

2.1 Class Overview

COMMON _ PPSE_STATUS is a class which include the response of the PPSE Selected.

2.2 Public constructor

COMMON _ PPSE_STATUS()

2.3 Public member

Field Name	Definition
Int retCode	The retCode of the PPSE selected
Byte[] SW = new byte[2]	The status of the PPSE selected
Int FCITemplateLen	The response length of the PPSE selected
Byte[] FCITemplate = new byte[256]	The response data of the PPSE selected

3 EMV_CAPK(class)

3.1 Class Overview

EMV_CAPK is a class which defined the struct data of the CAPK.

3.2 Public constructor

EMV_CAPK()

3.3 Public member

Field Name	Definition
byte[] RID = new byte[5]	CA public key RID
byte KeyID	CA public key index
byte HashInd	Hash algorithm index 1：SHA-1
byte ArithInd	Asymmetric algorithm index 1：RSA 4：SM

byte ModulLen	CA public key module length
byte[] Modul = new byte[248]	CA public key module
byte ExponentLen	CA public key exponent length. For SM algorithm, there is no need to set this field
byte[] Exponent = new byte[3]	CA public key exponent. For SM algorithm, there is no need to set this field
byte[] ExpDate = new byte[3]	CA public key expiry date in YYMMDD format
byte[] CheckSum = new byte[20]	CA public key checksum. The algorithm to calculate is as below: 1. Concatenate the CA public key RID(5bytes) 、 CA public key index(1byte)、CA public key module(n bytes) and CA public key exponent(n bytes) together, in the order mentioned above. 2. Apply SHA-1 hash algorithm on the concatenated data of the first step. The 20 bytes hash value is considered as the checksum of the CA public key.

4 CommonCB(interface)

4.1 Interface Overview

Since the kernel is independent of the hardware platform, so all the platform related operations are implemented by means of callback functions. The application should implement all the callback functions listed below.

For all the callback functions, the input parameters are provided by the kernel, the application only needs to read them; the output parameters should be set by the application and returned back to kernel. And the application should make sure the interface is conformed with the function definition such that proper return value is provided.

4.2 Interface

int	GetDateTime(byte[] DateTime)
int	ReadSN(byte[] SN)
int	GetUnknowTLV(int Tag, byte[] Data, int Len)

4.3 Interface detailed

4.3.1 GetDateTime

Prototype	int GetDateTime(byte[] DateTime)
Function	Get the terminal local date and time
Input	None

Output	DateTime	6 bytes date/time in BCD format YYMMDDhhmmss.
Returns	EMV_OK	Success
Note		

4.3.2 GetUnknowTLV

Prototype	int GetUnknowTLV(int Tag, byte[] Data, int Len)	
Function	Handles the Non-EMV defined tag	
Input	Tag	Tag name
	dat	Tag value
	len	Tag value length
Output	None	
Returns	-1	Success
Note	In current version of kernel, the application does not need to handle this.	

4.3.3 ReadSN

Prototype	int ReadSN(byte[] SN)	
Function	Get the terminal serial number	
Input	None	
Output	sn	Terminal serial number in ASCII code, ended

		with ‘\x0’.
Returns	EMV_OK	Success
Note		

5 Common(class)

5.1 Class Overview

include the some definitions, and the Common function used for the all kernel transaction running.

5.2 Public constructor

Common()

5.3 Public members

Field Name	Definition
public static final byte PEDICCARD = 1	Card reader interface definition, please refer to Common_SetIcCardType_Api.
public static final byte EXICCARD = 2	
public static final byte PEDPICCCARD = 3	
public static final byte EXPICCCARD = 4	
public static final byte PART_MATCH = 0	If the APP supports partial match
public static final byte FULL_MATCH = 1	
public static final int REFER_APPROVE = 1	Results of Voice Referral.
public static final int REFER_DENIAL = 2;	
public static final byte ONLINE_APPROVE = 0	Online results definition.
public static final byte ONLINE_FAILED = 1	
public static final byte ONLINE_REFER = 2	

public static final byte ONLINE_DENIAL = 3	
public static final byte ONLINE_ABORT = 4	
public static final int EMV_OK = 0	Success
public static final int ERR_EMVRSP = -2	Command response error
public static final int ERR_APPBLOCK = -3	Application already blocked
public static final int ERR_NOAPP = -4	No more candidate for selection
public static final int ERR_USERCANCEL = -5	User canceled
public static final int ERR_TIMEOUT = -6	Operation timeout
public static final int ERR_EMVDATA = -7	Error data returned by card
public static final int ERR_NOTACCEPT = -8	Transaction not accepted, should terminate
public static final int ERR_EMVDENIAL = -9	Transaction declined
public static final int ERR_KEYEXP = -10	CA public key already expired
public static final int ERR_NOPINPAD = -11	PINpad malfunction
public static final int ERR_NOPIN = -12	Cardholder bypassed the PIN entry process
public static final int ERR_CAPKCHECKSUM = -13	CA public key checksum error
public static final int ERR_NOTFOUND = -14	Data object not found
public static final int ERR_NODATA = -15	Data object does not have any value
public static final int ERR_OVERFLOW = -16	Memory overflow
public static final int ERR_ICCCMD = -21	Command exchange error
public static final int ERR_ICCBLOCK = -22	Card already blocked
public static final int ERR_USECONTACT = -25	Please use contact interface instead
public static final int ERR_APPEXP = -26	Application already expired

public static final int ERR_BLACKLIST = -27	Card is detected in exception file
public static final int ERR_UNSUPPORTED = -32	Function not supported
public static final int ERR_FILE = -101	Terminal file operation failed
public static final int ERR_PARAM = -102	Parameter error
public static final int ERR_PINBLOCK = -103	Cardholder PIN blocked
public static final int ERR_DATA_EXIST = -104	Duplicated data entry
public static final int ERR_AGAIN = -105	Please try again
public static final int ERR_ICCINSERTED = -111	IC card inserted
public static final int ERR_SEEPHONE = -112	Please refer to the consumer devices (such as ApplePay or similar NFC mobile devices). Normally it requires the cardholder to tap again
public static final int ERR_CLNOTALLOWED = -113	All the Contactless Applications are Not Allowed
public static final int ERR_SELECTNEXT = -114	Error occurs during the current application, one need to select the next one in the application candidate list
public static final int ERR_NOAMOUNT = -115	No amount

5.4 Public Methods

Static void	setCallback(CommonCB CB)
static String	GetVersion_Api()
static int	Init_Api()
static int	SetIcCardType_Api(byte Type, byte Slot)
static int	GetIcCardType_Api()
static int	SelectPPSE_Api(COMMON_PPSE_STATUS Status);
static int	SetTLV_Api(int Tag, byte[] Data, int len)
static int	GetTLV_Api(int Tag, byte[]Data, int[] len)
static int	GetTagAttr_Api(int Tag)
static void	ClearBlackList_Api()
static int	AddBlackList_Api(String cardNo, byte seq)
static int	DelBlackList_Api(String cardNo, byte seq)
static int	GetBlackList_Api(int index, byte[] blackPan, byte[] seq)
static int	AddCapk_Api(EMV_CAPK capk)
static int	GetCapk_Api(int Index, EMV_CAPK capk)
static int	SearchCapk_Api(EMV_CAPK capk, byte[] rid, byte KeyID)
static int	DelCapk_Api(byte KeyID, byte[] RID);
static int	CheckCapk_Api(byte[] KeyID, byte[] RID);
static void	ClearCapk_Api();
static int	AddIPKRevoke_Api(byte[] rid, byte capki, byte[] certserial)

static int	GetIPKRevoke_Api(int slotNo, byte[] rid, byte[] capki, byte[] certserial);
static int	DelIPKRevoke_Api(byte[] rid, byte capki, byte[] certserial);
static void	ClearIPKRevoke_Api();
static void	GetParam_Api(COMMON_TERMINAL_PARAM param);
static void	SetParam_Api(COMMON_TERMINAL_PARAM param)
static void	SaveParam_Api(COMMON_TERMINAL_PARAM param);

6 Public Methods detailed

6.1 Init_Api

Prototype	int Init_Api()	
Function	This function initializes the Common L2 Library.	
Input	None	
Output	None	
Returns	EMV_OK	Success
	ERR_KEYEXP	CA public key already expired
Note	This function should be called only ONCE before any other API is used	

6.2 GetVersion_Api

Prototype	String GetVersion_Api()	
Function	Retrieve the kernel version number	
Input	None	
Output	None	
Returns	”2.10”	kernel version string
Note	The format of kernel version is ”A.BC”.	

6.3 SetIcCardType_Api

Prototype	int SetIcCardType_Api(byte Type, byte Slot)	
Function	Select card interface	
Input	Type	PEDICCARD：Internal integrated contact interface EXICCARD：External contact interface PEDPICCCARD：Internal contactless interface EXPICCCARD：External contactless interface
	Slot	IC Card slot number，set to 0 by default
Output	None	
Returns	EMV_OK	Success
	ERR_PARAM	Invalid Mode value
Note	The application need to call this interface before performing IC card transactions.	

6.4 SelectPPSE_Api

Prototype	int SelectPPSE_Api(COMMON_PPSE_STATUS ComPPSEStatus)
Function	Select PPSE.
Input	None

Output	ComPPSEStatus	Response of PPSE selection related data structure, please refer to COMMON_PPSE_STATUS
Returns	EMV_OK	Success
	ERR_PARAM	Error Parameter.
Note	This function gives a list of directory entries for further application selection. The returned ComPPSEStatus should be delivered to each supported kernel one by one (by calling the related xxx_SelectApp_Api function), to determine which kernel should be used.	

6.5 GetParam_Api

Prototype	void GetParam_Api(COMMON_TERMINAL_PARAM Param)	
Function	Get the terminal parameters	
Input	None	
Output	Param	Terminal parameters Please refer to COMMON_TERMINAL_PARAM.
Returns	None	
Note		

6.6 SetParam_Api

Prototype	void SetParam_Api(COMMON_TERMINAL_PARAM Param)

Function	Set the terminal parameters	
Input	Param	Terminal parameters Please refer to COMMON_TERMINAL_PARAM.
Output	None	
Returns	None	
Note	This function only set the parameters in the memory, the parameters saved in file are not changed by this function. All the parameters will be reset to the saved value after reboot.	

6.7 SaveParam_Api

Prototype	void SaveParam_Api(const COMMON_TERMINAL_PARAM Param)	
Function	Set and save the terminal parameters	
Input	Param	Terminal parameters Please refer to COMMON_TERMINAL_PARAM.
Output	None	
Returns	None	
Note	This function is similar with Common_SetParam_Api. The difference is that the changed parameters are also saved into file after this function call.	

6.8 ClearBlackList_Api

Prototype	void ClearBlackList_Api(void)

Function	Delete all the records in terminal exception file.
Input	None
Output	None
Returns	None
Note	All the black list records saved in terminal will be removed.

6.9 AddBlackList_Api

Prototype	int AddBlackList_Api(byte[] cardNo, byte seq)	
Function	Add a record into the terminal exception file	
Input	cardNo	Black list card number, coded in ASCII format, ended with '\x0'.
	seq	Card sequence number
Output	None	
Returns	EMV_OK	Success
	ERR_OVERFLOW	Card number is too long(more than 20 numbers)
	ERR_EMV_FILE	File operation error
Note	If the card is already in the exception file, this function will return EMV_OK directly. There is no limitation on the maximum number of records in the terminal exception file, but note that it may take more time to finish a transaction if there are more records.	

6.10 DelBlackList_Api

Prototype	int DelBlackList_Api(byte[] cardNo, byte seq)	
Function	Delete a record in the terminal exception file	
Input	cardNo	Black list card number, coded in ASCII format, ended with '\x0'.
	seq	Card sequence number
Output	None	
Returns	EMV_OK	Success
	ERR_OVERFLOW	Card number is too long(more than 20 numbers)
	ERR_EMV_FILE	File operation error
	ERR_NOTFOUND	The specified record is not found.
Note		

6.11 GetBlackList_Api

Prototype	int GetBlackList_Api(int index, byte[] blackPan, byte[] seq)	
Function	Read a record from the terminal exception file	
Input	index	Record index, starting from 0
Output	blackPan	Black list card number, coded in ASCII format, ended with '\x0'.
	seq	Card sequence number
Returns	EMV_OK	Success
	ERR_FILE	File operation error

	ERR_NOTFOUND	The specified record is not found.
Note		

6.12 AddCapk_Api

Prototype	int AddCapk_Api(EMV_CAPK capk)	
Function	Add a CA public key	
Input	capk	CA public key structure, please refer to EMV_CAPK
Output	None	
Returns	EMV_OK	Success
	ERR_EMV_FILE	File operation error
	ERR_OVERFLOW	CA public key number exceeds upper limit
	ERR_CAPKCHECKSUM	CA public key checksum error
Note	The maximum number of CA public key supported in terminal. Each CA public key is identified by its RID and public key index. If the CA public key is already in the terminal, it will be replaced.	

6.13 GetCapk_Api

Prototype	int GetCapk_Api(int Index, EMV_CAPK capk)

Function	Retrieve a CA public key record	
Input	Index	Record index, starting from 0
Output	capk	CA public key structure, please refer to EMV_CAPK
Returns	EMV_OK	Success
	ERR_NOTFOUND	Specified index is not valid
Note		

6.14 SearchCapk_Api

Prototype	int Common_SearchCapk_Api(EMV_CAPK pCapk, byte[] rid, byte keyID)	
Function	Search for a CA public key record according to specified RID and CA public key index.	
Input	rid	CA public key RID, 5 bytes
	keyID	CA public key index
Output	pCapk	CA public key structure, please refer to EMV_CAPK
Returns	EMV_OK	Success
	ERR_NOTFOUND	Specified CA public key is not found
Note		

6.15 DelCapk_Api

Prototyp	int Common_DelCapk_Api(unsigned char KeyID, byte[] RID)

e		
Function	Delete a CA public key record according to specified RID and CA public key index.	
Input	RID	CA public key RID, 5 bytes
	KeyID	CA public key index
Output	None	
Returns	EMV_OK	Success
	ERR_NOTFOUND	Specified CA public key is not found
	ERR_EMV_FILE	File operation error
Note		

6.16 CheckCapk_Api

Prototype	int CheckCapk_Api(byte[] KeyID, byte[] RID)	
Function	Check the expiration date of all the CA public key stored in the terminal.	
Input	None	
Output	KeyID	The first expired CA public key index
	RID	The first expired CA public key RID, 5 bytes.
Returns	EMV_OK	All the CA public key is valid
	ERR_KEYEXP	There is at least ONE CA public key is expired
Note		

6.17 ClearCapk_Api

Prototype	void ClearCapk_Api(void)
Function	Clear all the CA public key stored in the terminal
Input	None
Output	None
Returns	None
Note	