



# **USER MANUAL**

## **Xpress CM100™**

### **API Reference Guide**

#### **V1.01**

**80099504-001-A**

**November 8, 2011**

### Software License Agreement

CAREFULLY READ ALL THE TERMS, CONDITIONS AND RESTRICTIONS OF THIS LICENSE AGREEMENT BEFORE USING OR INSTALLING THE SOFTWARE. YOUR USE OR INSTALLATION OF THE SOFTWARE PRESUMES YOUR AGREEMENT WITH AND ACCEPTANCE OF THE TERMS, CONDITIONS, AND RESTRICTIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS, CONDITIONS, AND RESTRICTIONS, PROMPTLY RETURN THE SOFTWARE AND RELATED DOCUMENTATION TO – ID TECH Support, 10721 Walker Street, Cypress, CA 90630.

#### TERMS, CONDITIONS AND RESTRICTIONS

ID TECH, Incorporated (the "Licensor") owns and has the right to distribute the described software and documentation, collectively referred to as the "Software".

LICENSE: Licensor grants you (the "Licensee") the right to use the Software in conjunction with ID TECH products.

LICENSEE MAY NOT COPY, MODIFY OR TRANSFER THE SOFTWARE IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT. Licensee may not decompile, disassemble, or in any other manner attempt to reverse engineer the Software. Licensee shall not tamper with, bypass, or alter any security features of the software or attempt to do so.

TRANSFER: Licensee may not transfer the Software or license to the Software to another party without prior written authorization of the Licensor. If Licensee transfers the Software without authorization, all rights granted under this Agreement are automatically terminated.

COPYRIGHT: The Software is copyrighted. Licensee may not copy the Software except to archive the Software or to load the Software for execution purposes. All other copies of the Software are in violation of this Agreement.

TERM: This Agreement is in effect as long as Licensee continues the use of the Software. The Licensor also reserves the right to terminate this Agreement if Licensee fails to comply with any of the terms, conditions, or restrictions contained herein. Should Licensor terminate this Agreement due to Licensee's failure to comply, Licensee agrees to return the Software to Licensor. Receipt of returned Software by the Licensor shall mark the termination.

## User Manual, Xpress CM100 API Guide

LIMITED WARRANTY: Licensor warrants to the Licensee that the disk(s) or other media on which the Software is recorded to be free from defects in material or workmanship under normal use. THE SOFTWARE IS PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Because of the diversity of conditions and PC hardware under which the Software may be used, Licensor does not warrant that the Software will meet Licensee specifications or that the operation of the Software will be uninterrupted or free of errors.

IN NO EVENT WILL LICENSOR BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE. Licensee's sole remedy in the event of a defect in material or workmanship is expressly limited to replacement of the Software disk(s) if applicable.

GOVERNING LAW: If any provision of this Agreement is found to be unlawful, void or unenforceable, that provision shall be removed from consideration under this Agreement and will not affect the enforceability of any of the remaining provisions. This Agreement shall be governed by the laws of the State of California and shall insure to the benefit of International Technologies & Systems Corporation (d/b/a ID TECH), its successors, or assigns.

ACKNOWLEDGMENT: LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS ALL OF ITS TERMS, CONDITIONS, AND RESTRICTIONS AND AGREES TO BE BOUND BY THEM. LICENSEE ALSO AGREES THAT THIS AGREEMENT SUPERSEDES ANY AND ALL, VERBAL AND WRITTEN, COMMUNICATIONS BETWEEN LICENSOR AND LICENSEE OR THEIR ASSIGNS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

QUESTIONS REGARDING THIS AGREEMENT SHOULD BE ADDRESSED IN WRITING TO ID TECH, INCORPORATED, ATTENTION: CUSTOMER SUPPORT, AT THE ABOVE ADDRESS OR E-MAILED TO: [support@idtechproducts.com](mailto:support@idtechproducts.com)

### **Information Provided**

**The information contained herein is provided to the user as a convenience. While every effort has been made to ensure accuracy, ID TECH is not responsible for damages that might occur because of errors or omissions, including any loss of profit or other commercial damage, nor for any infringements or patents or other rights of third parties that may result from its use. The specifications described herein were current at the time of publication, but are subject to change at any time without prior notice.**

## User Manual, Xpress CM100 API Guide

### Revision History

<b>Revision</b>	<b>Date</b>	<b>Description of Changes</b>	<b>By</b>
50	06/13/2011	Initial Draft	JW
51	07/18/2011	Modified for SDK v1.01	JW
A	11/08/2011	Revised Xpress_RetrievePublicKey description	JW

## Target Device:

Xpress CM100

## Description:

This API supports the Xpress CM100 device in USBHID and RS232 interface.

## Supported Platforms:

Microsoft Windows 2000, Windows XP, Vista

## DLL Usage:

 (Microsoft Visual C++ 6.0)

Add Xpress\_CM100Kit.lib to Project->Settings->Link->Object/library modules and include the header file Xpress\_CM100Kit.h, then call the DLL function directly. Please see Xpress\_CM100Test demo for more information.

## Command Summary

All commands supported are listed below.

Xpress\_OpenHid

Xpress\_OpenPort

Xpress\_SetBaud

Xpress\_SetParity

Xpress\_SetStopBit

Xpress\_Close

Xpress\_GetVersion

Xpress\_Beep

Xpress\_ControlLED

Xpress\_CancelCurrentTrans

Xpress\_GetTransData

Xpress\_GetTransResult

Xpress\_GetTransCVM

Xpress\_GetDateTime

Xpress\_SetDateTime

Xpress\_TerminalStartApp

Xpress\_RetrieveTerminalData

Xpress\_ReMoveTerminalData

Xpress\_SetTerminalData

Xpress\_SelectTerminalApp

Xpress\_RetrievePublicKey

Xpress\_RemovePublicKey

Xpress\_InvokePublicKey

Xpress\_RetrieveKeyCount

Xpress\_SetApplicationOff

Direct\_SearchCard

Direct\_SelectCard  
Direct\_DeleteCard  
Direct\_DeleteAllCard  
Direct\_GetCardCount  
Direct\_GetFreeSlots  
Direct\_GetCardCID  
Direct\_DisableCardCID  
Direct\_DisableCardNAD  
Direct\_SendCommand

## Function Description

<b>Function:</b>	Xpress_OpenHid	
<b>Description:</b>	Open USBHID device.	
<b>Format:</b>	BYTE Xpress_OpenHid ()	
<b>Parameter:</b>	None	
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_OpenHid ();	
<b>Function:</b>	Xpress_OpenPort	
<b>Description:</b>	Open RS232 device	
<b>Format:</b>	BYTE Xpress_OpenPort(int Comport,long Baud, char Parity, int Stop)	
<b>Parameter:</b>	Comport	Port number.
	Baud	Baud rate, default baud rate is 38400; please see Xpress_SetBaud.
	Parity	Parity check, default parity is None; please see Xpress_SetParity.
	Stop	Stop bit, default stop is 1; please see Xpress_SetStopBit.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Pay_OpenPort(1,38400, 'N', 1);	
<b>Function:</b>	Xpress_SetBaud	
<b>Description:</b>	Set the device's baudrate.	
<b>Format:</b>	BYTE Xpress_SetBaud(long Baud)	
<b>Parameter:</b>	Baud	The baudrate; valid value are 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600 or 115200.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetBaud(38400);	
<b>Function:</b>	Xpress_SetParity	
<b>Description:</b>	Set the device's parity style.	
<b>Format:</b>	BYTE Xpress_SetParity(char Parity)	
<b>Parameter:</b>	Parity	The device's parity:
		'N':None
		'O':Odd
		'E':Even
		'M':Mark 'S':Space
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetParity('N');	

## User Manual, Xpress CM100 API Guide

<b>Function:</b>	Xpress_SetStopBit	
<b>Description:</b>	Set the device's stop bit.	
<b>Format:</b>	BYTE Xpress_SetStopBit(int Stop)	
<b>Parameter:</b>	Stop	The stop bit; it is 1 or 2.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetStopBit(1);	
<b>Function:</b>	Xpress_Close	
<b>Description:</b>	Close device.	
<b>Format:</b>	bool Xpress_Close ()	
<b>Parameter:</b>	None	
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_Close ();	
<b>Function:</b>	Xpress_GetVersion	
<b>Description:</b>	Get the system's version.	
<b>Format:</b>	BYTE Xpress_GetVersion(int Type,char *Version, int *Length)	
<b>Parameter:</b>	Type	The type of version: 0x01:COMMAND_GET_VERSION_READER 0x02:COMMAND_GET_LEVEL1_VERSION 0x03:COMMAND_GET_VERSION_PAYPASS 0x04:COMMAND_GET_VERSION_VISA 0x05:COMMAND_GET_VERSION_EXPRESS 0x06:COMMAND_GET_VERSION_DISCOVER
	Version	Version string.
	Length	The length of Version string.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_GetVersion (0x01,Version, &length);	
<b>Function:</b>	Xpress_Beep	
<b>Description:</b>	Make the reader's beeper to beep for a designated time.	
<b>Format:</b>	BYTE Xpress_Beep(unsigned char Duration)	
<b>Parameter:</b>	Duration	the beeper's beep last time, the unit is 0.1 second.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_Beep (0x16);	
<b>Function:</b>	Xpress_ControlLED	
<b>Description:</b>	Control the system's led.	
<b>Format:</b>	BYTE Xpress_ControlLED(unsigned char f_Led)	



## User Manual, Xpress CM100 API Guide

<b>Parameter:</b>	f_Led	The led lighting map: 0x00: all leds off; 0x01: led1 on, others off; 0x02: led1&led2 on, others off; 0x03: all 4 leds on.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_ControlLED (0x01);	
<b>Function:</b>	Xpress_CancelCurrentTrans	
<b>Description:</b>	Cancel current transaction	
<b>Format:</b>	BYTE Xpress_CancelCurrentTrans()	
<b>Parameter:</b>	None	
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_CancelCurrentTrans();	
<b>Function:</b>	Xpress_GetTransData	
<b>Description:</b>	Get the transaction log data , use after Xpress_TerminalStartApp every time; For more information please see Xpress_CM100Test demo.	
<b>Format:</b>	BYTE Xpress_GetTransData(unsigned char *Type,unsigned char *Card_Data,unsigned int *Length)	
<b>Parameter:</b>	Type	The type of return data: MASTER_APPLICATION 0x01 VISA_APPLICATION 0x02 EXPRESS_APPLICATION 0x03 DISCOVER_APPLICATION 0x04
	Card_Data	The buffer of return data;For more data detail please see Appendix C.
	Length	The length of data.
<b>Return:</b>	Appendix A	
<b>Example:</b>	res = Xpress_TerminalStartApp(0x00,0x01,"1500"); res = Xpress_GetTransData(&type,rec,&len);	
<b>Function:</b>	Xpress_GetTransResult	
<b>Description:</b>	Get the last transaction result information; The reader will indicate to the terminal the outcome of its transaction processing by this command.	
<b>Format:</b>	BYTE Xpress_GetTransResult(unsigned char *Result)	
<b>Parameter:</b>	Result	The transaction result: TRANSACTION_TERMINATE 0x7E TRANSACTION_TERMINATE_TRY_ANOTHER

## User Manual, Xpress CM100 API Guide

		0x7D TRANSACTION_DECLINE 0x81 TRANSACTION_ACCEPT_OFFLINE 0x82 TRANSACTION_GO_ONLINE 0x84 TRANSACTION_COMPLETE 0x80
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_GetTransResult(&result);	
<b>Function:</b>	Xpress_GetTransCVM	
<b>Description:</b>	Get the last terminal verification CVM, this API is only for Paypass card; use after Xpress_GetTransData.	
<b>Format:</b>	BYTE Xpress_GetTransCVM(unsigned char *TransCVM)	
<b>Parameter:</b>	TransCVM	the last terminal verification CVM: 0x01:NoCVM 0x02:OnlinePin 0x03:Signaure 0x04:OfflinePin 0x05:OfflPinSign
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_GetTransCVM (&TransCVM);	
<b>Function:</b>	Xpress_GetDateTime	
<b>Description:</b>	Get the current date and time.	
<b>Format:</b>	BYTE Xpress_GetDateTime(char *DateTime, int *Length)	
<b>Parameter:</b>	DateTime	The date and time,the format is YYYYMMDDHHMMSS
	Length	The length of date and time.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_GetDateTime (DateTime,&length);	
<b>Function:</b>	Xpress_SetDateTime	
<b>Description:</b>	Set the date and time in the system.	
<b>Format:</b>	BYTE Xpress_SetDateTime(char *DateTime)	
<b>Parameter:</b>	DateTime	The date and time; the format must be YYYYMMDDHHMMSS.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetDateTime (“20110504102321”);	
<b>Function:</b>	Xpress_TerminalStartApp	
<b>Description:</b>	Instruct the reader to start to discover card	
<b>Format:</b>	BYTE Xpress_TerminalStartApp(unsigned char type, int Poll_Time,char *m_Amount)	

## User Manual, Xpress CM100 API Guide

<b>Parameter</b>	type	Indicates if the transaction is continuous or just for one time. 0x00: one-time transaction; 0x01: continuous transaction, and it will use the < TLV data > of this continuous command as transaction data.
	Poll_Time	Polling Card TimeOut, if the time is out,the interface will power off and this command need to be sent again to start another transaction. 0x00:keep polling untill reach card; Other is Poll_Time*2s.
	m_Amount	The amount data.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_TerminalStartApp (0x00,0x00,"1500");	
<b>Function:</b>	Xpress_RetrieveTerminalData	
<b>Description:</b>	Read the terminal data from the reader.	
<b>Format:</b>	BYTE Xpress_RetrieveTerminalData(int DataID,unsigned char AID,unsigned char *t_Data,int *length)	
<b>Parameter:</b>	DataID	Desired data element's Serial Number for Host to Identify ; DataID is from 1~43 excluding 36&40; For more information please see Appedix B.
	AID	Desired data belongs to and AID is 0x00~0x05, 0x00:No AID; 0x01:Visa AID; 0x02,0x03 and 0x04:Paypass AID; 0x05:Express AID; For more information please see Appedix B.
	t_Data	The buffer of Requested data element; For more information please see Appedix B.
	length	The length of Requested data element.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_RetrieveTerminalData(13,0x01,data,&length);	
<b>Function:</b>	Xpress_SetTerminalData	
<b>Description:</b>	Set terminal data.	
<b>Format:</b>	BYTE Xpress_SetTerminalData(int DataID,unsigned char AID,unsigned char *t_Data,int length)	
<b>Parameter:</b>	DataID	DataID is from 1~43 excluding 36&40; For more information please see Appedix B.
	AID	AID is 0x00~0x05:

## User Manual, Xpress CM100 API Guide

		<p>0x00:No AID          0x01:Visa AID          0x02,0x03 or 0x04:Paypass AID          0x05:Express AID          For more information please see Appedix B.</p>
	t_Data	The buffer for setting data; For more information please see Appedix B.
	length	The length of setting data,it must correspond with Appedix B;e.g.the length must be 6 if DataID is 4.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetTerminalData (1,0x00,t_Data,length);	
<b>Function:</b>	Xpress_ReMoveTerminalData	
<b>Description:</b>	Clear the terminal data	
<b>Format:</b>	BYTE Xpress_ReMoveTerminalData(int DataID,unsigned char AID)	
<b>Parameter:</b>	DataID	DataID is from1~43 excluding 36&40; For more information please see Appedix B.
	AID	AID is 0x00~0x05: 0x00:No AID 0x01:Visa AID 0x02,0x03 and 0x04:Paypass AID 0x05:Express AID For more information please see Appedix B.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_ReMoveTerminalData(1,0x00);	
<b>Function:</b>	Xpress_SelectTerminalApp	
<b>Description:</b>	Select the Reader's mode.	
<b>Format:</b>	BYTE Xpress_SelectTerminalApp(unsigned char *Type)	
<b>Parameter:</b>	Type	the reader's mode: 0x00: USER_MODE 0x01: MASTER_APPLICATION 0x02: VISA_APPLICATION 0x03: EXPRESS_APPLICATION 0x04: DISCOVER_APPLICATION
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SelectTerminalApp(0x01);	
<b>Function:</b>	Xpress_RetrievePublicKey	
<b>Description:</b>	Retrieve the terminal's saved public key of a special RID and its index.	
<b>Format:</b>	BYTE Xpress_RetrievePublicKey(unsigned char *m_RID,unsigned	

## User Manual, Xpress CM100 API Guide

	char m_Index,unsigned char* m_Key,int *length)	
<b>Parameter:</b>	m_RID	the applications's RID,it must be 5 bytes.
	m_Index	the key index.
	m_Key	The buffer for public key. RID(5 bytes)+Index(1 byte) + Hash_Indicator(1 byte) +Algorithm_Indicator(1 byte)+Exponent(4 bytes) +Key Module Length(1 byte)+ Key Hash(20 bytes) + Module Data(var)
	length	The length of public key.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_RetrievePublicKey (m_Rid,0x97,m_Key,&length);	
<b>Function:</b>	Xpress_RemovePublicKey	
<b>Description:</b>	Revoke the terminal's saved public key of a special RID and its index.	
<b>Format:</b>	BYTE Xpress_RemovePublicKey(unsigned char *m_RID,unsigned char m_Index)	
<b>Parameter:</b>	m_RID	the applications's RID,it must be 5 bytes.
	m_Index	The index key.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_RemovePublicKey (m_Rid,0x97);	
<b>Function:</b>	Xpress_InvokePublicKey	
<b>Description:</b>	Invoke a public key of a certain RID and its index in the terminal.	
<b>Format:</b>	BYTE Xpress_InvokePublicKey(unsigned char *m_RID,unsigned char m_Index,unsigned char *m_Exponent,unsigned char *m_Key,int length)	
<b>Parameter:</b>	m_RID	The applications's RID,it must be 5 bytes.
	m_Index	The index key.
	m_Exponent	The exponent data,it must be 4 bytes and add 0x00 in front if there is not 4 bytes.
	m_Key	The buffer for key.
	length	The length of key.
<b>Return:</b>	Appendix A	
<b>Example:</b>	<pre>int res = 0;     unsigned char m_Rid[5] = {0xA0,0x00,0x00,0x00,0x04};     unsigned char m_Index = 0xFA;     unsigned char m_Exponent[4] = {0x00,0x00,0x00,0x03};     unsigned char m_Value[144] = {0xA9,0x0F,0xCD,0x55,0xAA,0x2D,0x5D,0x99,0x63,0xE3,0x5E,0 xD0,0xF4,0x40,0x17,0x76,</pre>	

## User Manual, Xpress CM100 API Guide

	<pre> 0x99,0x83,0x2F,0x49,0xC6,0xBA,0xB1,0x5C,0xDA,0xE5,0x79,0x 4B,0xE9,0x3F,0x93,0x4D, 0x44,0x62,0xD5,0xD1,0x27,0x62,0xE4,0x8C,0x38,0xBA,0x83,0xD 8,0x44,0x5D,0xEA,0xA7, 0x41,0x95,0xA3,0x01,0xA1,0x02,0xB2,0xF1,0x14,0xEA,0xDA,0x 0D,0x18,0x0E,0xE5,0xE7, 0xA5,0xC7,0x3E,0x0C,0x4E,0x11,0xF6,0x7A,0x43,0xDD,0xAB,0 x5D,0x55,0x68,0x3B,0x14, 0x74,0xCC,0x06,0x27,0xF4,0x4B,0x8D,0x30,0x88,0xA4,0x92,0xF F,0xAA,0xDA,0xD4,0xF4, 0x24,0x22,0xD0,0xE7,0x01,0x35,0x36,0xC3,0xC4,0x9A,0xD3,0xD 0,0xFA,0xE9,0x64,0x59, 0xB0,0xF6,0xB1,0xB6,0x05,0x65,0x38,0xA3,0xD6,0xD4,0x46,0x4 0,0xF9,0x44,0x67,0xB1, 0x08,0x86,0x7D,0xEC,0x40,0xFA,0xAE,0xCD,0x74,0x0C,0x00,0x E2,0xB7,0xA8,0x85,0x2D};         res                                     = Xpress_InvokePublicKey(m_Rid,m_Index,m_Exponent,m_Value,14 4); </pre>	
<b>Function:</b>	Xpress_RetrieveKeyCount	
<b>Description:</b>	Retrieve the count of the terminal's saved public key of a special RID.	
<b>Format:</b>	BYTE Xpress_RetrieveKeyCount(unsigned char *m_RID,int *m_Count)	
<b>Parameter:</b>	m_RID	The special RID,it must be 5 bytes.
	m_Count	The count of public key for special RID.
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_RetrieveKeyCount (m_Rid,&m_Count);	
<b>Function:</b>	Xpress_SetApplicationOff	
<b>Description:</b>	Turn off application	
<b>Format:</b>	BYTE Xpress_SetApplicationOff()	
<b>Parameter:</b>	None	
<b>Return:</b>	Appendix A	
<b>Example:</b>	Xpress_SetApplicationOff();	
<b>Function:</b>	Direct_SearchCard	
<b>Description:</b>	Detect cards in the field	
<b>Format:</b>	BYTE Direct_SearchCard(unsigned char cType,unsigned char aType, int *CardCount,unsigned char *Card_Buf)	

## User Manual, Xpress CM100 API Guide

<b>Parameter:</b>	cType	Card type: 0x00 for type A and type B, 0x01 for type A and 0x02 for type B.
	aType	Application family identifier for type B. 0x00 for type A;
	CardCount	Detect card count;
	Card_Buf	Detect card data;the format is [<CardType><UIDLen><UID>][<CardType><UIDLen><UID>]... < CardType > is the card's type: 1 for type A and 2 for type B. <UIDLen> is the following UID's length. <UID> is the found card's UID (for type A card) or PUID (for type B card).
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_SearchCard(0x01,0x00,&CardCount,rec);	
<b>Function:</b>	Direct_SelectCard	
<b>Description:</b>	Select one card	
<b>Format:</b>	BYTE Direct_SelectCard(unsigned char cType,int UIDLen,unsigned char* UID,unsigned char* CardHandle)	
<b>Parameter:</b>	cType	the card's type: 0x01 for type A and 0x02 for type B.
	UIDLen	the length of the selected UID for type A or PUPI for type B
	UID	The UID of select Card.
	CardHandle	the selected card's handle, it will be used by other commands to communicate with a specified card
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_SelectCard(0x01,pthis->Card_Data[0].len,pthis->Card_Data[0].uid,&CardHandle)	
<b>Function:</b>	Direct_DeleteCard	
<b>Description:</b>	Delete one card from the system	
<b>Format:</b>	BYTE Direct_DeleteCard(unsigned char CardHandle)	
<b>Parameter:</b>	CardHandle	the Card's handle returned by Direct_SelectCard
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_DeleteCard(CardHandle);	
<b>Function:</b>	Direct_DeleteAllCard	
<b>Description:</b>	Delete all cards from the system	

## User Manual, Xpress CM100 API Guide

<b>Format:</b>	BYTE Direct_DeleteAllCard()	
<b>Parameter:</b>	None	
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_DeleteAllCard()	
<b>Function:</b>	Direct_GetCardCount	
<b>Description:</b>	Get the cards count using by the device system	
<b>Format:</b>	BYTE Direct_GetCardCount(unsigned char *CardCount)	
<b>Parameter:</b>	CardCount	Get the cards count
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_GetCardCount(&CardCount)	
<b>Function:</b>	Direct_GetFreeSlots	
<b>Description:</b>	Get the free slots available in the device system	
<b>Format:</b>	BYTE Direct_GetFreeSlots(unsigned char *FreeCount)	
<b>Parameter:</b>	FreeCount	Free slots available
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_GetFreeSlots(&FreeCount)	
<b>Function:</b>	Direct_GetCardCID	
<b>Description:</b>	Retrieve one card's current CID used in the device system	
<b>Format:</b>	BYTE Direct_GetCardCID(unsigned char CardHandle,unsigned char* CardCID)	
<b>Parameter:</b>	CardHandle	the Card's handle returned by Direct_SelectCard
	CardCID	Card's CID, it is between 0 and 14
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_GetCardCID(CardHandle,&CardCID)	
<b>Function:</b>	Direct_DisableCardCID	
<b>Description:</b>	Inform the device do not use card's CID in the system.	
<b>Format:</b>	BYTE Direct_DisableCardCID(unsigned char CardHandle)	
<b>Parameter:</b>	CardHandle	the Card's handle returned by Direct_SelectCard
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_DisableCardCID(CardHandle)	
<b>Function:</b>	Direct_DisableCardNAD	



## User Manual, Xpress CM100 API Guide

<b>Description:</b>	Inform the device do not use card's NAD in the system	
<b>Format:</b>	BYTE Direct_DisableCardNAD(unsigned char CardHandle)	
<b>Parameter:</b>	CardHandle	the Card's handle returned by Direct_SelectCard
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_DisableCardNAD(CardHandle)	
<b>Function:</b>	Direct_SendCommand	
<b>Description:</b>	Make the device to communicate with the card directly in the system	
<b>Format:</b>	BYTE Direct_SendCommand(unsigned char CardHandle,unsigned char* cBlock,int cLen,unsigned char* cData,int *dLen)	
<b>Parameter:</b>	CardHandle	the Card's handle returned by Direct_SelectCard
	cBlock	Input command
	cLen	The length of input command
	cData	Return data
	dLen	The length of return data
<b>Return:</b>	Appendix A	
<b>Example:</b>	Direct_SendCommand(CardHandle,Auth_Block,8,rec,&rLen)	

### Example for DLL call:

```
//include head file
#include "Xpress_CM100Kit.h"
//add Lib(Xpress_CM100Kit.lib):
Add Xpress_CM100Kit.lib to Project->Settings->Link->Object/library
//Call DLL functions using single-thread method:
// Xpress_OpenHid
    BYTE res = 0;
    res = Xpress_OpenHid();
// Xpress_Close
    BYTE res = 0;
    res = Xpress_Close();
// Xpress_GetVersion
    BYTE res = 0;
    char rec[128];
    int len = 0;
    m_EditVersion = "";
    int index = m_ConVersion.GetCurSel();
    res = Xpress_GetVersion(index+1,rec,&len);
    CString temp;
    if(res == 1)
    {
```

```
        for(int i = 0; i < len; i++)
            temp += rec[i];
    }
    else
        temp = "Get Version Fail";
    m_EditVersion = temp;
    UpdateData(false);
// Xpress_Beep
    BYTE res = 0;
    res = Xpress_Beep(0x10);
// Xpress_ControlLED
    BYTE res = Xpress_ControlLED (0x03);
// Xpress_CancelCurrentTrans
    BYTE res = 0;
    res = Xpress_CancelCurrentTrans();
    m_TransData = false;
// Xpress_GetDateTime
    BYTE res = 0;
    char buf[128];
    int length = 0;
    m_EditDate = "";
    res = Xpress_GetDateTime(buf,&length);
    CString str;
    if(res == 1)
    {
        for(int i = 0; i < length; i++)
            str += buf[i];
    }
    else
        str = "Get datetime fail";
    m_EditDate = str;
    UpdateData(false);
// Xpress_SetDateTime
    char buf[128];
    strcpy(buf,m_SetDate);
    res = Xpress_SetDateTime(buf);
    if(res == 1)
        MessageBox("Set DateTime success");
    else
        MessageBox("Set DateTime fail");
// Xpress_RetrieveTerminalData
    unsigned char data[512];
    int length = 0;
    int res = 0;
```

```
m_RData = "";
res = Xpress_RetrieveTerminalData(1,0x01,data,&length);
if(res == 1)
{
    CString temp;
    for(int i = 0; i < length; i++)
    {
        temp.Format("%02X ",data[i]);
        m_RData += temp;
    }
}
else
    m_RData = "Retrieve Data fail";
UpdateData(false);
// Xpress_ReMoveTerminalData
BYTE res = 0;
res = Xpress_ReMoveTerminalData(1,0x01);
// Xpress_SetTerminalData
unsigned char Data[5] = {0x60,0x00,0x00,0x00,0x00};
BYTE res = 0;
res = Xpress_SetTerminalData(13,0x01,Data,5);
// Call DLL functions using multi-threads methods:
// Xpress_TerminalStartApp
// Xpress_GetTransData (See Xpress_CM100 Test demo software for more
information)
//Xpress_GetTransResult
// Xpress_GetTransCVM
static UINT ThreadProc_Trans( LPVOID pParam )
{
    CXpress_CM100TestDlg* pthis = (CXpress_CM100TestDlg*)pParam;
    unsigned char rec[512];
    unsigned char type;
    unsigned int len;
    unsigned char Result;
    BYTE res = 0;
    ///////////////////////////////////
    res = Xpress_TerminalStartApp(0x00,0x00,"1500");
    if(res != 1)
    {
        pthis->m_TransData = false;
    }
    while(pthis->m_TransData)
    {
```

```
res = Xpress_GetTransData(&type,rec,&len);
if(res == 1)
{
    pthis->m_TansData = "";
    CString temp;
    if(type == 0x01)
        temp = "Card Type:Master Card\r\n";
    else if(type == 0x02)
        temp = "Card Type:Visa Card\r\n";
    else if(type == 0x03)
        temp = "Card Type:Express Card\r\n";
    else if(type == 0x04)
        temp = "Card Type:Discover Card\r\n";
    else
        temp = "Card Type:Not Known Card\r\n";
    pthis->m_TansData = temp;
    for(int i = 0; i < len; i++)
    {
        temp.Format("%02x ",rec[i]);
        pthis->m_TansData += temp;
    }
    res = Xpress_GetTransResult(&Result);
    if(res == 1)
    {
        temp.Format("\r\nTransResult:%02X",Result);
        pthis->m_TansData += temp;
    }
    res = Xpress_GetTransCVM(&Result);
    if(res == 1)
    {
        temp.Format("\r\nTransCVM:%02X",Result);
        pthis->m_TansData += temp;
    }
    pthis->SendMessage(WM_SWITCH_UPDATE, 0, 0);
    break;
}
Sleep(200);
}
//////////
return 0;
}
void CXpress_CM100TestDlg::OnGetdatbtn()
{
    // TODO: Add your control notification handler code here
```

```
m_TransData = true;  
  
AfxBeginThread(ThreadProc_Trans, this);  
}
```

**Appendix A: Return Value**

Return Value	Description
0	FAIL
1	SUCCESS
99	PARAMETER_ERR
100	ERROR_LOWOUTBUFFER
101	ERROR_CARD_NOT_FOUND
102	ERROR_COLLISION_CARD_EXIST
103	ERROR_TOOMANY_CARDS_EXIST
104	ERROR_SAVED_DATA_NOT_EXIST
105	ERROR_NO_DATA_AVAILABLE
106	ERROR_INVALID_CID_RETURNED
107	ERROR_INVALID_CARD_EXIST
108	ERROR_COMMAND_UNSUPPORTED
109	ERROR_COMMAND_PROCESS
110	ERROR_INVALID_COMMAND
111	ERROR_NOT_SUPPOERT
200	ERROR_HAS_OPENED
201	ERROR_NOT_OPENED

## Appendix B

### Note1:

AID = 0x00 (0x00, 0x00, 0x00, 0x00, 0x00, 0x00) No AID

AID = 0x01 (0xA0, 0x00, 0x00, 0x00, 0x03, 0x00) Visa AID

AID = 0x02 (0xA0, 0x00, 0x00, 0x00, 0x04, 0x10) Paypass AID

AID = 0x03 (0xA0, 0x00, 0x00, 0x00, 0x04, 0x30) Paypass AID

AID = 0x04 (0xB0, 0x12, 0x34, 0x56, 0x78, 0x00) Paypass AID

AID = 0x05 (0xA0, 0x00, 0x00, 0x00, 0x25, 0x01) Express AID

### Note2:

“Visa Terminal SetFlags” value: Byte 0~3, definition is as following, note that B1 viz.Byte0, is the first one comes in the command line.

- ✧ B1b1 (b1 viz.bit0) : "Status Check" ---see: "Visa Contactless Payment Specification: Req 5.52"
- ✧ B1b2: "Amount, Authorized of Zero Check" -----see: "Req 5.52"
- ✧ B1b3: "Reader Contactless Transaction Limit Check" ---see: "Req 5.52"
- ✧ B1b4: "Reader CVM Required Limit Check" -----see: "Req 5.52"
- ✧ B1b5: "Reader Contactless Floor Limit Check": Configurable to indicate whether this check is enabled or disabled. ---see: "Req 5.52"
- ✧ B1b6: "Terminal Floor Limit present", indicates if "Terminal Floor Limit '9f1b'" is present, ---see: "Req 5.36"
- ✧ B1b7 (b7 viz.bit6) : "Reader Contactless Floor Limit present", indicates if "Reader Contactless Floor Limit" is present. If it's present, FloorLimit\_Check must use "Reader Contactless Floor Limit" and ignore "terminal\_floor\_limit": ---see: "Req 5.36"
- ✧ B1b8: "Disable Online Currently" ---for "CLQ.K.001.02" in "Test Plan for Visa Contactless Payment Specification Version 2.1.1 qVSDC Contactless Readers"
- ✧ B2: RFU
- ✧ B3: (viz. Byte2) "Amount, Authorized of Zero Check: option(1) or (2)" ---see "Req 5.32 option 1 or 2"
- ✧ B4: (viz. Byte3) "MSD CVN17 enable(1) or disable(0)"

### Note3:

“Visa Terminal UpFlags” indicates the items need to transport to host after transaction finished. It's value has 4bytes: byte0~byte4, definition is as following, note that B1 viz.Byte0, is the first one comes in the command line.

- ✧ B1: RFU
- ✧ B2b1(b1 viz.bit0): 0X9F6E, Form Factor Indicator (FFI)
- ✧ B2b2: 0X9F7C, Customer Exclusive Data (CED)
- ✧ B2b3: 0X9F27, Cryptogram Information Data
- ✧ B2b4: 0X9F02, Amount, Authorized (Numeric)
- ✧ B2b5: 0X9F03, Amount, Other (Numeric)
- ✧ B2b6: 0X9F33, Terminal Capabilities
- ✧ B2b7~8: RFU

## User Manual, Xpress CM100 API Guide

- ✧ B3b1: 0X9A , Transaction Date
- ✧ B3b2: 0X9C , Transaction Type
- ✧ B3b3: 0X9F10, Issuer Application Data
- ✧ B3b4: 0X9F1A, Terminal Country Code
- ✧ B3b5: 0X9F26, Application Cryptogram
- ✧ B3b6: 0X9F36, Application Transaction Counter (ATC)
- ✧ B3b7: 0X9F37, Unpredictable Number
- ✧ B3b8: 0X9F5D, Available Offline Spending Amount (AOSA)
- ✧ B4b1: 0X57 , Track 2 Equivalent Data
- ✧ B4b2: 0X5A , Application Primary Account Number (PAN)
- ✧ B4b3: 0X5F20, Cardholder Name
- ✧ B4b4: 0X5F24, Application Expiration Date
- ✧ B4b5: 0X5F2A, Transaction Currency Code
- ✧ B4b6: 0X5F34, Application Primary Account Number Sequence Number (PSN)
- ✧ B4b7: 0X82 , Application Interchange Profile (AIP)
- ✧ B4b8: 0X95 , Terminal Verification Results (TVR)
- ✧ B5b1: TRACK1\_0x1f11
- ✧ B5b2~4: RFU
- ✧ B5b5: TRACK2\_0x1f12
- ✧ B5b6~8: RFU

### Note4:

“Express Terminal SetFlags” value: Byte 0~3, definition is as following, note that B1 viz. Byte 0, is the first one comes in the command line:

- ✧ B1b1(viz.bit0): support another interface flag
- ✧ B1b2: Terminal sends GENAC command requesting ARQC with CDA signature
- ✧ B1b3: Full online supported
- ✧ B1b4: Optimised Processing Supported
- B1b5: Online Capable, but now cannot go online

### Note5:

"Unpredictable Number Range" value: Byte 0~3,

EXAMPLE:

- ✧ {0x00,0x00,0x00,0x3C}: means the range is 0~60;
- ✧ {0x00,0x00,0x01,0x60}: means the range is 0~316;
- ✧ {0x00,0x00,0x04,0xb0}: means the range is 0~1200;

The default range is 0 to 0x3C, which is also the minimum range.

**Please bear in mind the maximum range as per the Amex spec is 0 to 1200**

### Terminal Data List

DataID	Name	Tag	Length (bytes)	AID
1	IFD Serial Number	0x9F1E	8	0x00
2	Terminal Country Code	0x9F1A	2	0x00



## User Manual, Xpress CM100 API Guide

3	Amount Authorized (Binary)	0x8100	4	0x00
4	Amount Authorized (Numeric) Equal to Amount Authorized (Binary) Range(0~0x4294967295)	0x9F02	6	0x00
5	Amount Other (Binary)	0x9F04	4	0x00
6	Amount Other (Numeric) Equal to Amount Other (Binary) Range(0~0x4294967295)	0x9F03	6	0x00
7	Transaction Category Code	0x9F53	1	0x00
8	Transaction Currency Code	0x5F2A	2	0x00
9	Transaction Currency Exponent	0x5F36	1	0x00
10	Terminal Date	0x9A00	3	0x00
11	Terminal Time	0x9F21	3	0x00
12	Transaction Type	0x9C00	1	0x00
13	Additional Terminal Capabilities	0x9F40	5	0x01,0x02,0x03,0x04 or 0x05
14	Application Version Number	0x9F09	2	0x01,0x02,0x03,0x04 or 0x05
15	Default UDOL	0x0000	3	0x01,0x02,0x03,0x04 or 0x05
16	MagStripe Application Version Number	0x9f6d	2	0x01,0x02,0x03,0x04 or 0x05
17	Merchant Category Code	0x9F15	2	0x01,0x02,0x03,0x04 or 0x05
18	PayPass - Mag Stripe Indicator	0x0000	1	0x01,0x02,0x03,0x04 or 0x05
19	Terminal Action Codes-default	0x0000	5	0x01,0x02,0x03,0x04 or 0x05
20	Terminal Action Codes-denial	0x0000	5	0x01,0x02,0x03,0x04 or 0x05
21	Terminal Action Codes-online	0x0000	5	0x01,0x02,0x03,0x04 or 0x05
22	Terminal Type	0x9F35	1	0x01,0x02,0x03,0x04 or 0x05
23	Terminal Capabilities - No CVM Required	0x0000	3	0x01,0x02,0x03,0x04 or 0x05

## User Manual, Xpress CM100 API Guide

24	Terminal Capabilities - CVM Required	0x0000	3	0x01,0x02,0x03,0x04 or 0x05
25	Terminal Contactless Transaction Limit Range(0~0x4294967295)	0x0000	6	0x01,0x02,0x03,0x04 or 0x05
26	Terminal Contactless Floor Limit Range(0~0x4294967295)	0x0000	6	0x01,0x02,0x03,0x04 or 0x05
27	Terminal CVM Required Limit Range(0~0x4294967295)	0x0000	6	0x01,0x02,0x03,0x04 or 0x05
28	Merchant Custom Data	0x9f7c	20	0x01,0x02,0x03,0x04 or 0x05
29	Terminal Floor Limit	0x9f1b	4	0x01,0x02,0x03,0x04 or 0x05
30	Merchant Name and Location	0x9f4e	Var	0x00
31	Merchant Identifier	0x9f16	15	0x00
32	Visa Terminal Transaction Qualifiers(TTQ)	0x9f66	4	0x01,0x02,0x03,0x04 or 0x05
33	Visa Terminal SetFlags(See Note2)	0x0000	4	0x01,0x02,0x03,0x04 or 0x05
34	Visa Terminal UpFlags(see Note3)	0x0000	5	0x01,0x02,0x03,0x04 or 0x05
35	Terminal Capabilities	0x9f33	3	0x01,0x02,0x03,0x04 or 0x05
37	Target Percentage for Random Selection	0x0000	1	0x01,0x02,0x03,0x04 or 0x05
38	Maximum Target Percentage for Biased Random Selection	0x0000	1	0x01,0x02,0x03,0x04 or 0x05
39	Threshold Value for Biased Random Selection	0x0000	4	0x01,0x02,0x03,0x04 or 0x05
41	Expresspay Terminal Capabilities	0x9f6d	1	0x01,0x02,0x03,0x04 or 0x05
42	Express Terminal SetFlags(see Note4)	0x0000	4	0x01,0x02,0x03,0x04 or 0x05
43	Unpredictable Number Range(see Note5)	0x0000	4	0x01,0x02,0x03,0x04 or 0x05

### Terminal Data Default Value

DataID	PayPass Default	VISA Default	American Default
--------	-----------------	--------------	------------------

## User Manual, Xpress CM100 API Guide

1	None		
2	0x00 0x56		
3	None		
4	Set by Xpress_TerminalStartApp()		
5	0x00		
6	0x00		
7	0x00		
8	0x09 0x78		
9	0x02		
10	None		
11			
12	0x00		
13	0x60 0x00 0x00 0x00 0x00	0x20 0x00 0x00 0x00 0x01	0x60 0x00 0x00 0x00 0x00
14	0x00 0x02	None	0x00 0x01
15	0x9F 0x6A 0x04	None	None
16	0x00 0x01	None	None
17	0x07 0x42	None	0x07 0x42
18	0x01	None	None
19	0x00 0x00 0x00 0x00 0x00	None	0xc4 0x00 0x00 0x00 0x00
20	0x00 0x00 0x00 0x00 0x00	None	0xc4 0x00 0x00 0x00 0x00
21	0x00 0x00 0x00 0x00 0x00	None	0xc4 0x00 0x00 0x00 0x00
22	0x25	None	0x25
23	0x00 0x08 0x88	None	None
24	0x00 0x28 0x88	None	None
25	0x00 0x00 0x00 0x03 0x00 0x00	0x00 0x00 0x00 0x03 0x00 0x00	0x00 0x00 0x00 0x01 0x00 0x00
26	0x00 0x00 0x00 0x01 0x00 0x00	0x00 0x00 0x00 0x01 0x00 0x00	0x00 0x00 0x00 0x01 0x00 0x00
27	0x00 0x00 0x00 0x01 0x00 0x00	0x00 0x00 0x00 0x60 0x00 0x00	0x00 0x00 0x00 0x01 0x00 0x00
28	0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10 0x12 0x13 0x14 0x15	None	None
29	None	0x00 0x00 0x27 0x10	0x00 0x00 0x27 0x10
30	None	0x49 0x44 0x20 0x54	None

## User Manual, Xpress CM100 API Guide

		0x45 0x43 0x48 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x43 0x41 0x20 0x55 0x53 0x41	
31	None	0x49 0x44 0x30 0x54 0x45 0x43 0x48 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	None
32	None	0xA0 0x00 0x00 0x00	None
33	None	0x5E 0x00 0x01 0x01	None
34	None	0xFF 0xFF 0xFF 0xFF 0xFF	None
35	None	None	0x00 0x28 0x88
37	None	None	0x3C Range(0x00~0x63)
38	None	None	0x3C Range (DataID:37's Value~0x63)
39	None	None	0x00 0x00 0x00 Range(0~floorlimit)
41	None	None	0x80
42	None	None	0x0A 0x00 0x00 0x00
43	None	None	0x00 0x00 0x00 0x3C Range(0~1200)

### Appendix C: Data Record Detail

#### C-1: Data Record Detail for Amex- Mag \_Stripe and Discover- Mag \_Stripe

Tag	Name
0x1f11	Magstripe Track1
0x1f12	Magstripe Track2

#### C-2: Data Record Detail for Amex- EMV

Tag	Name
0x82	Application Interchange Profile
0x9F36	Application Transaction Counter
0x9F26	ARQC
0x9F27	CID
0x9F34	CVM Results
0x9F1E	IFD Serial Number

## User Manual, Xpress CM100 API Guide

0x9F10	Issuer Application Data
0x9F33	Terminal Capabilities
0x9F35	Terminal Type
0x95	TVR
0x9F37	Unpredictable Number
0x9F02	Amount, Authorised
0x9F03	Amount, Other
0x5F25	Application Effective Date
0x5F24	Application Expiration Date
0x5A	Application PAN
0x5F34	Application PAN Sequence Number
0x9F15	Merchant Category Code
0x9F1A	Terminal Country Code
0x57	Track 2 Equivalent Data
0x5F2A	Transaction Currency Code
0x9A	Transaction Date
0x9F21	Transaction Time
0x9C	Transaction Type

### C-3: Data Record Detail for Paypass- M/Chip

Tag	Name
0x57	Track 2 Equivalent Data
0x9F6E	PayPass Third Party Data
0x84	DF Name
0x50	Application Label
0x9F12	Application Preferred Name
0x9F11	Issuer Code Table Index
0x9F26	AC
0x9F27	CID
0x9F10	IAD
0x9F36	ATC
0x95	TVR 6
0x9F37	UN
0x5F2A	Transaction Currency Code
0x9C	Transaction Type
0x9f33	Terminal Capabilities
0x9A	Transaction Date
0x9F02	Transaction Amount
0x9F1A	Terminal Country Code
0x9F34	CVM Results
0x82	AIP
0x5A	Application PAN
0x5F34	Application PAN Sequence Number

## User Manual, Xpress CM100 API Guide

0x5F24	Application Expiration Date
--------	-----------------------------

### C-4: Data Record Detail for Paypass- Mag\_Stripe

Tag	Name
0x9F6B	Track 2 Data
0x56	Track 1 Data
0x1f11	DDCARD,TRACK1
0x1f12	DDCARD,TRACK2
0x9F6E	PayPass Third Party Data
0x84	DF Name
0x50	Application Label
0x9F12	Application Preferred Name
0x9F11	Issuer Code Table Index

### C-5: Data Record Detail for Visa

Tag	Name
1f11	Track1
1f12	Track2
1f13	POS Entry Mode
1f14	Terminal Entry Capability
0X9F6E	Form Factor Indicator (FFI)
0X9F7C	Customer Exclusive Data (CED)
0X9F27	Cryptogram Information Data
0X9F02	Amount, Authorized (Numeric)
0X9F03	Amount, Other (Numeric)
0X9F33	Terminal Capabilities
0X9A	Transaction Date
0X9C	Transaction Type
0X9F10	Issuer Application Data
0X9F1A	Terminal Country Code
0X9F26	Application Cryptogram
0X9F36	Application Transaction Counter (ATC)
0X9F37	Unpredictable Number
0X9F5D	Available Offline Spending Amount (AOSA)
0X57	Track 2 Equivalent Data
0X5A	Application Primary Account Number (PAN)
0X5F20	Cardholder Name
0X5F24	Application Expiration Date
0X5F2A	Transaction Currency Code
0X5F34	Application Primary Account Number Sequence Number (PSN)
0X82	Application Interchange Profile (AIP)
0X95	Terminal Verification Results (TVR)
0X9f66	Terminal Transaction Qualifiers