

CVReader SDK

CVReader SDK is software development kit worked with CV3XXX, CV5XXX and CV6XXX product series for second development. It includes API, Host Demo, user manual, USB driver and program example. It's developed by C++builder developing tool on Windows Operating system and GCC developing tool on Linux OS, compatible to VC,VB/VB.NET,JAVA, C++,C#.NET, DELPHI etc.

Features

- ✚ SDK is available with some special benefits for customer's development
- ✚ Stronger and powerful informative API to save your time and cost for development (See "CVReader API reference directory" in Appendix for details)
- ✚ Examples for good reference and understanding, using developing tools like Delphi, C++ Builder, C#.NET, VC,VB/VB.NET etc.
- ✚ Comprehensive platform to provide flexibility by customer side:
Operating System: Linux and Windows. Support Win 98,Win ME, Win 2000, Win 2003, Win 2003 x64, Win XP, Win XP x64, Win Vista, Win Vista x64, Win 2008, Win 2008 x64, Win 2008 R2 x64, Win 7, Win 7 x64, Win8
- ✚ Host Demo, a software testing tool to ease and simply development
- ✚ One SDK for all our readers (CV3XXX, CV5XXX, CV6XXX) to avoid development cost for product upgrade

Selective SDK

CVReader SDK	ISO14443A/B, Mifare, ISO15693
CVReader Finger SDK	ISO14443A/B, Mifare, ISO15693, Fingerprint
CVReader_DESFire_Finger SDK	ISO14443A/B, Mifare, ISO15693, Mifare Plus, Desfire EV1 and Fingerprint

Appendix of CVReader API Reference Directory

CVReader API Reference Directory

<u>1</u>	<u>INTRODUCTION</u>	14
<u>1.1</u>	<u>Corresponding reader</u>	14
<u>1.2</u>	<u>serial port communication introduction</u>	14
1.2.1	<u>CV_SetCommunicationType(byte aType)</u>	14
1.2.2	<u>SearchAndOpenUSBPorts (void)</u>	14
1.2.3	<u>SearchUSBReadersCallBack(TOnSearchUSBReaders aOnSearchUSBReaders)</u>	14
1.2.4	<u>UsingUSBPort(int iUSBPort)</u>	15
1.2.5	<u>CloseUSBPorts(void)</u>	15
1.2.6	<u>CV_UDPRun(int SvrPort)</u>	15
1.2.7	<u>CV_UDPStop()</u>	15
1.2.8	<u>CV_SetRemoteIPPort(unsigned char *ip,int port)</u>	15
1.2.9	<u>important notice</u>	16
<u>1.2.10</u>	<u>Single serial port working mode</u>	17
<u>1.2.11</u>	<u>Multi serial port working mode.</u>	17
<u>2</u>	<u>API FUNCTIONS</u>	19

2.1	<u>Configuration, I/O commands</u>	19
2.1.1	<u>int GetVersionAPI(char *VersionAPI)</u>	19
2.1.2	<u>int OpenComm(char *Com, int Baudrate)//.. single serial port working mode</u>	19
2.1.3	<u>int CloseComm (Void)</u>	19
2.1.4	<u>int SetExtPort(int DeviceAddress,unsigned char Ports)</u>	19
2.1.5	<u>int GetPort (int DeviceAddress, unsigned char *status)</u>	20
2.1.6	<u>int ActiveLED (int DeviceAddress, unsigned char NumLED, unsigned char ontime, unsigned char cycle)</u>	20
2.1.7	<u>int SetLED (int DeviceAddress, unsigned char LEDState)</u>	21
2.1.8	<u>int ActiveBuzzer (int DeviceAddress, unsigned char mode, unsigned char *pattern)</u>	21
2.1.9	<u>int RF_Field (int DeviceAddress, unsigned char time)</u>	22
2.1.10	<u>int SetWiegandStatus(int DeviceAddress,unsigned char status)</u>	22
2.1.11	<u>int ActiveWiegandMode(int DeviceAddress,unsigned char status)</u>	23
2.1.12	<u>int CreateCommPort(int SerialNUM,unsigned char *CommID,int Baudrate) //..multi serial port working mode [note]:the function will only available on serial port communication(232 communication)</u>	23
2.1.13	<u>int ReaderC_EXT(int DeviceAddress,byte iTime)[Controlling Relay if reader have]</u>	25
2.2	<u>Auto-Read mode APIs</u>	25
	<u>Note: before you call the following APIs, reader must works in wiegant mode, or reader will not read card automatically. You can chose only one method to support auto-read mode: calling API ReadOriginalData() by yourself or the following APIs.</u>	
2.2.1	<u>int SetOnAutoRead(TOnAutoRead aEventFun)</u>	25
2.2.2	<u>int AutoRead_Run()</u>	25
2.2.3	<u>int AutoRead_Stop()</u>	25
2.2.4	<u>int IsAutoRead ()</u>	26
2.3	<u>ISO14443-A Command</u>	27
2.3.1	<u>int MF_Request (int DeviceAddress, unsigned char mode, unsigned char *ATQ)</u>	27
2.3.2	<u>int MF_Anticoll (int DeviceAddress, unsigned char *UID, unsigned char &Collision)</u>	27
2.3.3	<u>int MF_Anticoll2 (int DeviceAddress, unsigned char *UID, unsigned char &Collision)</u>	28
2.3.4	<u>int MF_Anticoll3 (int DeviceAddress, unsigned char *UID, unsigned char &Collision)</u>	28
2.3.5	<u>int MF_Select(int Device Address, unsigned char *UID)</u>	28
2.3.6	<u>int MF_Select2(int Device Address, unsigned char *UID)</u>	29
2.3.7	<u>int MF_Select3(int Device Address, unsigned char *UID)</u>	29
2.3.8	<u>int MF_Halt(int Device Address, unsigned char mode)</u>	30
2.3.9	<u>int SLE_Generic(int DeviceAddress,unsigned char CRC_Flag,unsigned char &length,unsigned char *buffer)</u>	30
2.3.10	<u>int SLE_GenericEx (int DeviceAddress,unsigned char CRC_Flag,unsigned char TimeOut,unsigned char &length,unsigned char *buffer)</u>	30
2.4	<u>Mifare function</u>	32
2.4.1	<u>int MF_Auth(int DeviceAddress, unsigned char KeyAB, unsigned char *snr, unsigned char add_blk)</u>	32
2.4.2	<u>int MF_Read (int DeviceAddress,unsigned char add_blk, unsigned char num_blk, unsigned char *buffer);</u>	32

<u>2.4.3</u>	<u>int MF Write(int DeviceAddress,unsigned char add_blk, unsigned char num_blk, unsigned char *buffer)</u>	32
<u>2.4.4</u>	<u>int MF Transfer(int DeviceAddress, unsigned char add_blk)</u>	33
<u>2.4.5</u>	<u>int MF Increment(int DeviceAddress, unsigned char add_blk, int value)</u>	33
<u>2.4.6</u>	<u>int MF Decrement(int DeviceAddress, unsigned char add_blk, int value)</u>	33
<u>2.4.7</u>	<u>int MF Restore(int DeviceAddress, unsigned char add_blk)</u>	34
<u>2.4.8</u>	<u>int MF InitValue(int DeviceAddress, unsigned char add_blk, int value)</u>	34
<u>2.4.9</u>	<u>int MF ReadValue(int DeviceAddress, unsigned char add_blk, int *value)</u>	34
<u>2.4.10</u>	<u>int MF LoadKey(int DeviceAddress, unsigned char *Key)</u>	34
<u>2.4.1</u>	<u>int MF StoreKeyToEE(int DeviceAddress, unsigned char KeyAB, unsigned char Sector, unsigned char *Key)</u>	35
<u>2.4.2</u>	<u>int MF LoadKeyFromEF(int DeviceAddress, unsigned char KeyAB, unsigned char Sector)</u>	35
2.3.4	Mifare High Level Functions	36
<u>2.4.3</u>	<u>int MF HLRead (int DeviceAddress, unsigned char mode, unsigned char add_blk, unsigned char num_blk, unsigned char *snr, unsigned char *buffer);</u>	36
<u>2.4.4</u>	<u>int MF HLWrite (int DeviceAddress, unsigned char mode, unsigned char add_blk, unsigned char num_blk, unsigned char *snr, unsigned char *buffer);</u>	36
<u>2.4.5</u>	<u>int MF HLInitVal (int DeviceAddress, unsigned char mode, unsigned char sect_num, unsigned char *snr, int value);</u>	37
<u>2.4.6</u>	<u>int MF HLInc (int DeviceAddress, unsigned char mode, unsigned char sect_num, unsigned char *snr, int *value)</u>	38
<u>2.4.7</u>	<u>int MF HLDec (int DeviceAddress, unsigned char mode, unsigned char sect_num, unsigned char *snr, int *value)</u>	38
<u>2.4.8</u>	<u>int MF HLRequest (int DeviceAddress, unsigned char mode, int &length, unsigned char *UID)</u>	39
2.5	ISO14443 B Commands	40
<u>2.5.1</u>	<u>int Generic_B(int DeviceAddress,unsigned char &length, unsigned char *buffer) ;</u>	40
<u>2.5.2</u>	<u>int Generic_BEx (int DeviceAddress,unsigned char CRC_Flag,unsigned char TimeOut, unsigned char &length,unsigned char *buffer);</u>	40
2.6	ISO-15693 Interface Commands	41
<u>2.6.1</u>	<u>int CMD_ISO15693_Generic(int DeviceAddress,unsigned char length,unsigned char *databuffer)</u>	41
<u>2.6.2</u>	<u>int CMD_ISO15693_Inventory(int DeviceAddress,unsigned char flags,unsigned char AFI,unsigned char masklength ,unsigned char *maskvalue,unsigned char*databuffer</u>	41
<u>2.6.3</u>	<u>int CMD_ISO15693_Stay_Quiet(int DeviceAddress,unsigned char flags,unsigned char *UID)</u>	42
<u>2.6.4</u>	<u>int stdcall CMD_ISO15693_Read(int DeviceAddress,unsigned char flags,unsigned char firstblock, unsigned char Number,unsigned char*UID,unsigned char*databuffer)</u>	42
<u>2.6.5</u>	<u>Int CMD_ISO15693_Write(int DeviceAddress,unsigned char flags,unsigned char Firstblock, unsigned char Number,unsigned char*UID,unsigned char*databuffer)</u>	43
<u>2.6.6</u>	<u>Int CMD_ISO15693_Lock_Block(int DeviceAddress,unsigned char flags,unsigned char block,unsigned char*UID)</u>	43
<u>2.6.7</u>	<u>int CMD_ISO15693_Select(int DeviceAddress,unsigned char flags,unsigned char*UID)</u>	44

<u>2.6.8</u>	<u>int CMD ISO15693 Reset To Ready(int DeviceAddress,unsigned char flags,unsigned char*UID)</u>	44
<u>2.6.9</u>	<u>int CMD ISO15693 Write_AFI(int DeviceAddress,unsigned char flags, unsigned char AFI, unsigned char*UID)</u>	45
<u>2.6.10</u>	<u>int CMD ISO15693 Lock_AFI(int DeviceAddress,unsigned char flags,unsigned char*UID)</u>	45
<u>2.6.11</u>	<u>Int CMD ISO15693 Write_DSfid(int DeviceAddress,unsigned char flags,unsigned char DSfid,unsigned char*UID)</u>	46
<u>2.6.12</u>	<u>int CMD ISO15693 Lock_DSfid(int DeviceAddress,unsigned char flags,unsigned char*UID)</u>	46
<u>2.6.13</u>	<u>Int CMD ISO15693 GET System Information(int DeviceAddress,unsigned char flags,unsigned char*UID,unsigned char *databuffer)</u>	47
<u>2.6.14</u>	<u>int CMD ISO15693 GetMultipleBlockSecurity(int DeviceAddress,unsigned char flags,unsigned char Firstblock,unsigned char Number,unsigned char *UID,unsigned char *databuffer)</u>	47
<u>2.6.15</u>	<u>int CMD Myd_read(int DeviceAddress,unsigned char flags,unsigned char block,unsigned char*UID,unsigned char*databuffer)</u>	48
<u>2.6.16</u>	<u>int CMD Myd_write(int DeviceAddress,unsigned char flags,unsigned char block,unsigned char*UID,unsigned char*databuffer)</u>	48
<u>2.6.17</u>	<u>int CMD ISO15693 Inventory16(int DeviceAddress,unsigned char Flags, unsigned char AFI,int *CardNum,unsigned char*Buffer)</u>	49

2.7 Misc. Commands **51**

<u>2.7.1</u>	<u>int SetFirmwareBaudrate(int DeviceAddress, unsigned char Baudrate)</u>	51
<u>2.7.2</u>	<u>int SetDeviceAddress (int DeviceAddress, unsigned char &newAddress</u>	51
<u>2.7.3</u>	<u>int GetVersionNum (int DeviceAddress, char *VersionNUM)</u>	52
<u>2.7.4</u>	<u>int GetSerialNum (int DeviceAddress, int &CurrentAddress, char *SerialNUM)</u>	52
<u>2.7.5</u>	<u>int GetUserInfo (int DeviceAddress, char *UserInfo)</u>	52
<u>2.7.6</u>	<u>int SetUserInfo (int DeviceAddress, char *UserInfo)</u>	53
<u>2.7.7</u>	<u>int SetWiegandStatus(int DeviceAddress,unsigned char status)</u>	53
<u>2.7.8</u>	<u>int ActiveWiegandMode(int DeviceAddress,unsigned char status)</u>	53
<u>2.7.9</u>	<u>int WiegandMode (int DeviceAddress,unsigned char*data)</u>	54
<u>2.7.10</u>	<u>int LcdDisplayLogo(int DeviceAddress,unsigned char*data);</u>	55
<u>2.7.11</u>	<u>int LcdDisplay(int DeviceAddress,unsigned char address,unsigned char length, char *Dstring);</u>	55
<u>2.7.12</u>	<u>GetKey(int DeviceAddress,unsigned char *keybuffer);</u>	55
<u>2.7.13</u>	<u>int ReadChar(unsigned char *byte)</u>	56
<u>2.7.14</u>	<u>int ReadOriginalData (unsigned long iTimeOut, unsigned long iReadLength, unsigned char *pDataBur, unsigned long *iSuccessLen)</u>	56

2.8 UltraLight Card Commands **56**

<u>2.8.1</u>	<u>REQA command use ISO14443A REQA command</u>	56
<u>2.8.2</u>	<u>WAKE-UP command use ISO14443A WAKE-UP command</u>	56
<u>2.8.3</u>	<u>HALT command use ISO14443A HALT command</u>	56
<u>2.8.4</u>	<u>int MF_UL_AnticollL1(int DeviceAddress,unsigned char *UID_Part1,unsigned char *BCC1)</u>	56
<u>2.8.5</u>	<u>int MF_UL_SelectL1(int DeviceAddress,unsigned char *UID_Part1,unsigned char BCC1)</u>	57

<u>2.8.6</u>	<u>int MF_UL_AnticollL2(int DeviceAddress,unsigned char *UID_Part2,unsigned char *BCC2)</u>	57
<u>2.8.7</u>	<u>int MF_UL_SelectL2(int DeviceAddress,unsigned char *UID_Part2,unsigned char BCC2)</u>	57
<u>2.8.8</u>	<u>int MF_UL_Read(int DeviceAddress,int StartPage , unsigned char *FromData_16Bytes)</u>	57
<u>2.8.9</u>	<u>int MF_UL_Write(int DeviceAddress,int PageNum , unsigned char *ToData_4Bytes)</u>	57

2.9 DESFire EV1 Card Commands 58

The DESFire EV1 Card are: MF31CD40, MF31CD21, MF31CD41, MF31CD8158

Some of the following APIs are invalid to MF31CD40, will be indicated. 58

<u>2.9.1</u>	<u>REQA command use ISO14443A REQA command</u>	58
<u>2.9.2</u>	<u>WAKE-UP command use ISO14443A WAKE-UP command</u>	58
<u>2.9.3</u>	<u>int MF_DF_AnticollL1 (int DeviceAddress,unsigned char *UID_Part1,unsigned char *BCC1)</u>	58
<u>2.9.4</u>	<u>int MF_DF_SelectL1(int DeviceAddress,unsigned char *UID_Part1,unsigned char BCC1)</u>	58
<u>2.9.5</u>	<u>int MF_DF_AnticollL2(int DeviceAddress,unsigned char *UID_Part2,unsigned char *BCC2)</u>	58
<u>2.9.6</u>	<u>int MF_DF_SelectL2(int DeviceAddress,unsigned char *UID_Part2,unsigned char BCC2)</u>	59
<u>2.9.7</u>	<u>MF_DF_RATS(int DeviceAddress,byte FSDI=8,byte CID=0)</u>	59
<u>2.9.8</u>	<u>MF_DF_RATSEx(int DeviceAddress,unsigned char CID,unsigned char FSDI,unsigned char *ATS)</u>	59
<u>2.9.9</u>	<u>MF_DF_PPS(int DeviceAddress,byte CID=0,byte DSI=0,byte DRI=0)</u>	60
<u>2.9.10</u>	<u>MF_DF_Authenticate(int DeviceAddress, unsigned char KeyNo, unsigned char *Key)</u>	60
	<u>The API has no CID parameter, it will operate on the current active card, for other information, please refer to its alternative API MF_DF_AuthenticateEx(),</u>	60
<u>2.9.11</u>	<u>MF_DF_AuthenticateEx(int DeviceAddress, unsigned char CID, unsigned char KeyNo, unsigned char *Key)</u>	60
<u>2.9.12</u>	<u>MF_DF_AuthenticateISO(int DeviceAddress, unsigned char CID, unsigned char KeyNo, unsigned char *Key)</u>	61
<u>2.9.13</u>	<u>MF_DF_AuthenticateAES(int DeviceAddress, unsigned char CID, unsigned char KeyNo, unsigned char *Key)</u>	62
<u>2.9.14</u>	<u>MF_DF_ChangeKeySettings(int DeviceAddress,byte MasterKeySettings=0x0F)</u>	63
	<u>The API has no CID parameter, it will operate on the current active card, for other information, please refer to its alternative API MF_DF_ChangeKeySettingsEx (),</u>	63
<u>2.9.15</u>	<u>MF_DF_ChangeKeySettingsEx(int DeviceAddress,unsigned char CID,unsigned char MasterKeySettings=0x0F)</u>	63
<u>2.9.16</u>	<u>MF_DF_GetKeySettings(int DeviceAddress,byte *MasterKeySettings,byte *MaxKey)</u>	65
	<u>The API has no CID parameter, it will operate on the current active card, for other information, please refer to its alternative API MF_DF_GetKeySettingsEx ().</u>	65
<u>2.9.17</u>	<u>MF_DF_GetKeySettingsEx(int DeviceAddress,unsigned char CID,unsigned char *MasterKeySettings,unsigned char *MaxKey,unsigned char *CryptoType)</u>	65
<u>2.9.18</u>	<u>MF_DF_ChangeKey(int DeviceAddress,byte KeyNo, byte *new_Key, byte *old_Key)</u>	66
<u>2.9.19</u>	<u>MF_DF_ChangeKeyEx(int DeviceAddress,unsigned char CID,unsigned char KeyNo, unsigned char AuthenticateType,unsigned char KeyVersion, unsigned char *new_Key ,unsigned char *old_Key)</u>	66
<u>2.9.20</u>	<u>MF_DF_GetKeyVersion(int DeviceAddress,byte KeyNo,byte *KeyVersion)</u>	66

<u>2.9.21 MF DF GetKeyVersionEx(int DeviceAddress,unsigned char CID,unsigned char KeyNo,unsigned char *KeyVersion)</u>	67
<u>2.9.22 MF DF CreateApplication(int DeviceAddress,int AID, byte MasterKeySettings=0x0F, byte NumOfKeys=0x0E)</u>	67
<u>2.9.23 MF DF CreateApplicationEx(int DeviceAddress,unsigned char CID,int AID,unsigned char MasterKeySettings, unsigned char NumOfKeys,unsigned char CryptoMethod)</u>	67
<u>2.9.24 MF DF DeleteApplication(int DeviceAddress,int AID);</u>	68
<u>2.9.25 MF DF DeleteApplicationEx(int DeviceAddress,unsigned char CID,int AID)</u>	68
<u>2.9.26 MF DF GetApplicationIDs(int DeviceAddress,int AIDs27[])</u>	68
<u>2.9.27 MF DF GetApplicationIDsEx(int DeviceAddress,unsigned char CID,int AIDs27[])</u>	68
<u>2.9.28 MF DF SelectApplication(int DeviceAddress,int AID)</u>	69
<u>2.9.29 MF DF SelectApplicationEx(int DeviceAddress, unsigned char CID, int AID)</u>	69
<u>2.9.30 MF DF FormatPICC(int DeviceAddress)</u>	69
<u>2.9.31 MF DF FormatPICCEX(int DeviceAddress, unsigned char CID)</u>	69
<u>2.9.32 MF DF GetVersion(int DeviceAddress, unsigned char VerInfo[28])</u>	70
<u>2.9.33 MF DF GetVersionEx(int DeviceAddress, unsigned char CID,unsigned char VerInfo[28])</u>	70
<u>2.9.34 MF DF GetFreeMem(int DeviceAddress,unsigned char CID,unsigned long *pFreeMemSize)</u>	71
<u>2.9.35 MF DF GetCardUID(int DeviceAddress,unsigned char CID,unsigned char *UID)</u>	71
<u>2.9.36 MF DF SetConfiguration(int DeviceAddress,unsigned char CID,unsigned char ConfigurationType, unsigned char *pConfigurationData,unsigned char ConfigurationDataLength)</u>	71
<u>2.9.37 MF DF GetFileIDs(int DeviceAddress,int FileIDs[16])</u>	72
<u>2.9.38 MF DF GetFileIDsEx(int DeviceAddress,unsigned char CID,unsigned char *FileIDs,unsigned char *FileNumber)</u>	72
<u>2.9.39 MF DF GetFileSettings(int DeviceAddress, unsigned char FileNo, unsigned char FileSettings[])</u>	72
<u>2.9.40 MF DF GetFileSettingsEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char FileSettings[])</u>	72
<u>2.9.41 MF DF ChangeFileSettings(int DeviceAddress, unsigned char FileNo, unsigned char CommSet,unsigned short AccessRights, int Is_Rights_Free)</u>	74
<u>2.9.42 MF DF ChangeFileSettingsEx(int DeviceAddress,unsigned char CID,unsigned char FileNo, unsigned char CommSet,unsigned short AccessRights,int Is_Rights_Free)</u>	74
<u>2.9.43 MF DF CreateStdDataFile(int DeviceAddress, unsigned char FileNo, unsigned char CommSet, unsigned short AccessRights, int FileSize)</u>	75
<u>2.9.44 MF DF CreateStdDataFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet, unsigned short AccessRights,int FileSize)</u>	75
<u>2.9.45 MF DF CreateBackupDataFile(int DeviceAddress, unsigned char FileNo, unsigned char CommSet, unsigned short AccessRights, int FileSize)</u>	76
<u>2.9.46 MF DF CreateBackupDataFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,unsigned short AccessRights,int FileSize)</u>	76
<u>2.9.47 MF DF CreateValueFile(int DeviceAddress,byte FileNo, unsigned char CommSet,unsigned short AccessRights, int LowerLimit,int UpperLimit,int Value, unsigned char LimitedCreditEnabled)</u>	76
<u>2.9.48 MF DF CreateValueFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,unsigned short AccessRights,int LowerLimit,int UpperLimit,int Value,unsigned char LimitedCreditEnabled)</u>	77

<u>2.9.49 MF DF CreateLinearRecordFile(int DeviceAddress,byte FileNo, unsigned char CommSet,unsigned short AccessRights, int RecordSize, int MaxNumRecords)</u>	77
<u>2.9.50 MF DF CreateLinearRecordFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,unsigned short AccessRights,int RecordSize,int MaxNumRecords)</u>	77
<u>2.9.51 MF DF CreateCyclicRecordFile(int DeviceAddress,byte FileNo, unsigned char CommSet,unsigned short AccessRights, int RecordSize, int MaxNumRecords)</u>	78
<u>2.9.52 MF DF CreateCyclicRecordFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,unsigned short AccessRights,int RecordSize,int MaxNumRecords)</u>	78
<u>2.9.53 MF DF DeleteFile(int DeviceAddress, unsigned char FileNo)</u>	79
<u>2.9.54 MF DF DeleteFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo)</u>	79
<u>2.9.55 MF DF ReadData(int DeviceAddress,byte FileNo, int Offset, int Length, unsigned char CommSet, unsigned char DataRead[255],int &ReadLen)</u>	79
<u>2.9.56 MF DF ReadDataEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,int Offset,int Length,unsigned char CommSet,unsigned char *DataRead,int &ReadLen)</u>	79
<u>2.9.57 MF DF WriteData(int DeviceAddress, unsigned char FileNo, int Offset,int Length, unsigned char CommSet, byte *DataToWrite)</u>	80
<u>2.9.58 MF DF WriteDataEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,int Offset,int Length, unsigned char CommSet,unsigned char *DataToWrite)</u>	80
<u>2.9.59 MF DF GetValue(int DeviceAddress, unsigned char FileNo, unsigned char CommSet, int *Value)</u>	81
<u>2.9.60 MF DF GetValueEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,int *Value)</u>	81
<u>2.9.61 MF DF Credit(int DeviceAddress, byte FileNo, unsigned char CommSet, const int incValue)</u>	81
<u>2.9.62 MF DF CreditEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,const unsigned long incValue)</u>	81
<u>2.9.63 MF DF Debit(int DeviceAddress, unsigned char FileNo,byte CommSet,const int decValue)</u>	82
<u>2.9.64 MF DF DebitEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,const unsigned long decValue)</u>	82
<u>2.9.65 MF DF LimitedCredit(int DeviceAddress, unsigned char FileNo, byte CommSet, const int incValue)</u>	83
<u>2.9.66 MF DF LimitedCreditEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,unsigned char CommSet,const unsigned long incValue)</u>	83
<u>2.9.67 MF DF WriteRecord(int DeviceAddress, unsigned char FileNo,int Offset, int Length, unsigned char CommSet, unsigned char *DataToWrite)</u>	83
<u>2.9.68 MF DF WriteRecordEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,int Offset,int Length,unsigned char CommSet,unsigned char *DataToWrite)</u>	84
<u>2.9.69 MF DF ReadRecords(int DeviceAddress,unsigned char FileNo,int Offset,int RecNumberToRead,int RecSize,unsigned char CommSet,unsigned char *DataRead,int &ReadLen)</u>	84
<u>2.9.70 MF DF ReadRecordsEx(int DeviceAddress,unsigned char CID,unsigned char FileNo,int Offset,int RecNumberToRead,int RecSize,unsigned char CommSet,unsigned char *DataRead,int &ReadLen)</u>	84
<u>2.9.71 MF DF ClearRecordFile(int DeviceAddress, unsigned char FileNo)</u>	85
<u>2.9.72 MF DF ClearRecordFileEx(int DeviceAddress,unsigned char CID,unsigned char FileNo)</u>	85
<u>2.9.73 MF DF CommitTransaction(int DeviceAddress)</u>	86
<u>2.9.74 MF DF CommitTransactionEx(int DeviceAddress,unsigned char CID)</u>	86

<u>2.9.75</u>	<u>MF DF AbortTransaction(int DeviceAddress)</u>	86
<u>2.9.76</u>	<u>MF DF AbortTransactionEx(int DeviceAddress,unsigned char CID)</u>	86
<u>2.9.77</u>	<u>MF DF SaveDESFire4Key(int DeviceAddress, int Index, int KeyNo,unsigned char Key[16])</u>	87
<u>2.9.78</u>	<u>MF DF SaveDESFire4ParameterForRead (int DeviceAddress, int AppNo, int FileNo, int Offset, int ReadLen, int KeyIndex)</u>	87
<u>2.9.79</u>	<u>MF DF ReadDESFire4UID(int DeviceAddress,unsigned char DESFireUID[7])</u>	87
<u>2.9.80</u>	<u>MF DF ReadDESFire4File (int DeviceAddress, int AppNo, int FileNo, int Offset, int ReadLen, int KeyIndex, int &FileType, int &Value, unsigned char *DataBuf, int &ReturnLen)</u>	87
<u>2.9.81</u>	<u>MF DF DESELECT(int DeviceAddress)</u>	88
<u>2.9.82</u>	<u>MF DF DESELECTEx(int DeviceAddress,unsigned char CID)</u>	88
2.10	<u>DESFire EV1 Card Commands</u>	88
	<u>The following APIs is valid for CV3XXXXA, CV5X2A-X-XX</u>	88
<u>2.10.1</u>	<u>REQA command use ISO14443A REQA command</u>	88
<u>2.10.2</u>	<u>WAKE-UP command use ISO14443A WAKE-UP command</u>	88
<u>2.10.3</u>	<u>Anticoll command use ISO14443A MF Anticoll command</u>	89
<u>2.10.4</u>	<u>Select command use ISO14443A MF Select command</u>	89
<u>2.10.5</u>	<u>Anticol2 command use ISO14443A MF Anticol2 command</u>	89
<u>2.10.6</u>	<u>SelectL2 command use ISO14443A MF SelectL2 command</u>	89
<u>2.10.7</u>	<u>RATS(int DeviceAddress,unsigned char CID,unsigned char FSDI,unsigned char *ATS,unsigned int *ATSLen)</u>	89
<u>2.10.8</u>	<u>PPS(int DeviceAddress,unsigned char CID, unsigned char DSI, unsigned char DRI)</u>	89
<u>2.10.9</u>	<u>DESELECT(int DeviceAddress,unsigned char CID)</u>	90
<u>2.10.10</u>	<u>MF DFEV1 Authenticate(int DeviceAddress,unsigned char CID,unsigned char CryptoType,unsigned char KeyNo, unsigned char *Key)</u>	90
<u>2.10.11</u>	<u>MF DFEV1 ChangeKeySettings(int DeviceAddress,unsigned char CID,unsigned char MasterKeySettings)</u>	91
<u>2.10.12</u>	<u>MF DFEV1 ChangeKeySettings(int DeviceAddress,unsigned char CID,unsigned char MasterKeySettings)</u>	93
<u>2.10.13</u>	<u>MF DFEV1 ChangeKey(int DeviceAddress,unsigned char CID,unsigned char KeyNo, unsigned char CryptoType, unsigned char KeyVersion, unsigned char *NewKey, unsigned char *OldKey)</u>	93
<u>2.10.14</u>	<u>MF DFEV1 GetKeyVersion(int DeviceAddress,unsigned char CID, unsigned char KeyNo,unsigned char *KeyVersion)</u>	94
<u>2.10.15</u>	<u>MF DFEV1 CreateApplication(int DeviceAddress,unsigned char CID, int AID,unsigned char MasterKeySettings, unsigned char KeyNum, unsigned char CryptoType)</u>	94
<u>2.10.16</u>	<u>MF DFEV1 DeleteApplication(int DeviceAddress,unsigned char CID, int AID)</u>	95
<u>2.10.17</u>	<u>MF DFEV1 GetApplicationIDs(int DeviceAddress,unsigned char CID, int *AIDs, int *AIDNum)</u>	95
<u>2.10.18</u>	<u>MF DFEV1 SelectApplication(int DeviceAddress,unsigned char CID, int AID)</u>	95
<u>2.10.19</u>	<u>MF DFEV1 FormatPICC(int DeviceAddress,unsigned char CID)</u>	96
<u>2.10.20</u>	<u>MF DFEV1 GetVersion(int DeviceAddress,unsigned char CID, unsigned char *VersionInfo,unsigned int *VersionInfoLen)</u>	96
<u>2.10.21</u>	<u>MF DFEV1 GetFreeMem(int DeviceAddress,unsigned char CID, int *RemainMemSize)</u>	97
<u>2.10.22</u>	<u>MF DFEV1 GetCardUID(int DeviceAddress,unsigned char CID, unsigned char *UID,unsigned int *UIDLen)</u>	97

<u>2.10.23 MF DFEV1 SetConfiguration(int DeviceAddress,unsigned char CID, unsigned char ConfigType, unsigned char *ConfigData,unsigned int ConfigDataLen)</u>	97
<u>2.10.24 MF DFEV1 GetFileIDs(int DeviceAddress,unsigned char CID, unsigned char *FileIDs,unsigned int *FileIDsLen)</u>	98
<u>2.10.25 MF DFEV1 GetFileSettings(int DeviceAddress,unsigned char CID,unsigned char FileID, unsigned char *FileSettings)</u>	98
<u>2.10.26 MF DFEV1 ChangeFileSettings(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet,unsigned short AccessRights,bool Is_Rights_Free)</u>	100
<u>2.10.27 MF DFEV1 CreateDataFile(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char FileType, unsigned char CmtSet,unsigned short AccRights,unsigned int CreateSize)</u>	101
<u>2.10.28 MF DFEV1 CreateValueFile(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet, unsigned short AccRights,unsigned int LLimit,unsigned int ULimit,unsigned int InitVal,unsigned char LimitCredit)</u>	101
<u>2.10.29 MF DFEV1 CreateRecordFile(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char FileType,unsigned char CmtSet, unsigned short AccRights,unsigned int SingleRecordSize,unsigned int RecordNumber)</u>	102
<u>2.10.30 MF DFEV1 DeleteFile(int DeviceAddress,unsigned char CID, unsigned char FileID)</u>	102
<u>2.10.31 MF DFEV1 ReadData(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet, unsigned int Offset,unsigned int ReadLength,unsigned char *ReturnData,unsigned int *ReturnDataLen)</u>	103
<u>2.10.32 MF DFEV1 WriteData(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet, unsigned int Offset,unsigned int WriteDataLength,unsigned char *WriteData)</u>	103
<u>2.10.33 MF DFEV1 GetValue(int DeviceAddress,unsigned char CID, unsigned char FileID,unsigned char CmtSet, int *RetVal)</u>	104
<u>2.10.34 MF DFEV1 Credit(int DeviceAddress,unsigned char CID, unsigned char FileID, unsigned char CmtSet,unsigned int IncValue)</u>	104
<u>2.10.35 F DFEV1 Debit(int DeviceAddress,unsigned char CID, unsigned char FileID, unsigned char CmtSet,unsigned int DecValue)</u>	105
<u>2.10.36 MF DFEV1 LimitedCredit(int DeviceAddress,unsigned char CID, unsigned char FileID, unsigned char CmtSet,unsigned int IncValue)</u>	106
<u>2.10.37 MF DFEV1 WriteRecord(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet, unsigned int Offset,unsigned int RecordLength,unsigned char *RecordData)</u>	106
<u>2.10.38 MF DFEV1 ReadRecord(int DeviceAddress,unsigned char CID,unsigned char FileID,unsigned char CmtSet, unsigned int Offset,unsigned int ReadNumber,unsigned int RecSize,unsigned char *ReturnData,unsigned int *ReturnDataLen)</u>	107
<u>2.10.39 MF DFEV1 ClearRecordFile(int DeviceAddress,unsigned char CID,unsigned char FileID)</u>	107
<u>2.10.40 MF DFEV1 CommitTransaction(int DeviceAddress,unsigned char CID)</u>	108
<u>2.10.41 MF DFEV1 AbortTransaction(int DeviceAddress,unsigned char CID)</u>	108
<u>2.10.42 MF DFEV1 KeySetting(int DeviceAddress,unsigned char KeyNo,unsigned char CryptoType,unsigned char *Key)</u>	109
<u>2.10.43 MF DFEV1 FileSetting(int DeviceAddress,int AID,unsigned char KeyNo,unsigned char FileID,unsigned char FileType, unsigned char CmtSet,unsigned int Offset,unsigned int Length,unsigned int RecSize)</u>	109
<u>2.10.44 MF DFEV1HL Read(int DeviceAddress,unsigned char CID,int AID,unsigned char KeyIndex, unsigned char FileID,unsigned char FileType,unsigned char CmtSet,unsigned int Offset,unsigned int Length, unsigned int RecSize,unsigned char *Data,unsigned int *ReadLen)</u>	109

<u>2.10.45</u>	<u>MF_DFEV1HL_Write(int DeviceAddress,unsigned char CID,int AID,unsigned char KeyIndex, unsigned char FileID,unsigned char FileType,unsigned char CmtSet,unsigned int Offset,unsigned int Length, unsigned char *Data)</u>	110
<u>2.10.46</u>	<u>MF_DFEV1HL_ClearReads (int DeviceAddress,unsigned char CID,int AID,unsigned char KeyIndex, unsigned char FileID)</u>	110
<u>2.10.47</u>	<u>MF_DFEV1HL_GetValue(int DeviceAddress,unsigned char CID,int AID,unsigned char KeyIndex, unsigned char FileID,unsigned char CmtSet, int *RetVal)</u>	111
<u>2.10.48</u>	<u>MF_DFEV1HL_Credit(int DeviceAddress,unsigned char CID,int AID,unsigned char KeyIndex, unsigned char FileID,unsigned char CmtSet, unsigned int IncValue,bool IsLimited)</u>	111
<u>2.10.49</u>	<u>MF_DFEV1HL_Debit(int DeviceAddress,unsigned char CID, int AID,unsigned char KeyIndex, unsigned char FileID,unsigned char CmtSet, unsigned int DecValue)</u>	111
2.11	<u>Fingerprint Commands</u>	112
<u>2.11.1</u>	<u>CV64GetMessage(unsigned long MsgID,char *MsgStr)</u>	112
<u>2.11.2</u>	<u>CV64SetMsgCallBack(CV64X0_MSGCALLBACK_TYPE MsgCallBack)</u>	112
<u>2.11.3</u>	<u>CV64Enroll(int DeviceAddress, unsigned char *pTemplate, unsigned long &pTemplateLength, unsigned long dwTimeOut=10000)</u>	112
<u>2.11.4</u>	<u>CV64Verify (int DeviceAddress, unsigned char *pTemplate, unsigned long &pTemplateLength, unsigned long dwTimeOut=10000)</u>	113
2.12	<u>Mifare Plus Card Commands</u>	113
<u>2.12.1</u>	<u>REQA command use ISO14443A REQA command</u>	113
<u>2.12.2</u>	<u>WAKE-UP command use ISO14443A WAKE-UP command</u>	113
<u>2.12.3</u>	<u>int MF_PS_Anticoll1(int DeviceAddress,unsigned char *UID_Part1,unsigned char *BCC1)</u>	113
<u>2.12.4</u>	<u>int MF_PS_SelectL1(int DeviceAddress,unsigned char *UID_Part1,unsigned char BCC1)</u>	113
<u>2.12.5</u>	<u>int MF_PS_Anticoll2(int DeviceAddress,unsigned char *UID_Part2,unsigned char *BCC2)</u>	114
<u>2.12.6</u>	<u>int MF_PS_SelectL2(int DeviceAddress,unsigned char *UID_Part2,unsigned char BCC2)</u>	114
<u>2.12.7</u>	<u>MF_PS_RATSEx(int DeviceAddress,unsigned char CID,unsigned char FSDI,unsigned char *ATS)</u>	114
<u>2.12.8</u>	<u>MF_FS_PPS(int DeviceAddress,byte CID=0,byte DSI=0,byte DRI=0)</u>	114
<u>2.12.9</u>	<u>MF_PS_DESELECTEx(int DeviceAddress,unsigned char CID)</u>	115
<u>2.12.10</u>	<u>MF_PS_WritePerso(int DeviceAddress,unsigned char CID,unsigned short BNr,unsigned char *Data)</u>	115
<u>2.12.11</u>	<u>MF_PS_CommitPerso(int DeviceAddress,unsigned char CID)</u>	116
<u>2.12.12</u>	<u>MF_PS_SL1AESAuth(int DeviceAddress,unsigned char *AESKey)</u>	116
<u>2.12.13</u>	<u>MF_PS_SL2AESAuth(int DeviceAddress,unsigned short BNr,unsigned char *AESKey,unsigned char *MFKey,unsigned char *MFSessionKey)</u>	117
<u>2.12.14</u>	<u>MF_PS_FirstAuth(int DeviceAddress,unsigned char CID,unsigned short BNr,unsigned char *AESKey)</u>	117
<u>2.12.15</u>	<u>MF_PS_FollowAuth(int DeviceAddress,unsigned char CID,bool IsPlusX,unsigned short BNr,unsigned char *AESKey)</u>	117
<u>2.12.16</u>	<u>MF_PS_SL3ResetAuth(int DeviceAddress,unsigned char CID)</u>	118

<u>2.12.17</u>	<u>MF PS SL3Read(int DeviceAddress,unsigned char CID,bool IsPlusX,unsigned short BNr,unsigned char BlockNum,bool MACC,bool Plaintext,bool MACR,unsigned char *DataRead,unsigned char *DataReadLen)</u>	118
<u>2.12.18</u>	<u>MF PS SL3Write(int DeviceAddress,unsigned char CID,bool IsPlusX,unsigned short BNr,bool Plaintext,bool MACR,unsigned char *DataToWrite,unsigned char DataToWriteLen)</u>	119
<u>2.12.19</u>	<u>MF PS SL3InitValue(int DeviceAddress,unsigned char CID,unsigned short BNr,bool Plaintext,bool MACR,int Value)</u>	119
<u>2.12.20</u>	<u>MF PS SL3ReadValue(int DeviceAddress,unsigned char CID,unsigned short BNr,bool MACC,bool Plaintext,bool MACR,int *Value)</u>	119
<u>2.12.21</u>	<u>MF PS SL3Increment(int DeviceAddress,unsigned char CID,unsigned short BNr,bool MACR,unsigned long Value)</u>	120
<u>2.12.22</u>	<u>MF PS SL3Decrement(int DeviceAddress,unsigned char CID,unsigned short BNr,bool MACR,unsigned long Value)</u>	120
<u>2.12.23</u>	<u>MF PS SL3Transfer(int DeviceAddress,unsigned char CID,unsigned short BNr,bool MACR)</u>	121
<u>2.12.24</u>	<u>MF PS SL3Restore(int DeviceAddress,unsigned char CID,unsigned short BNr,bool MACR)</u>	121
<u>2.12.25</u>	<u>MF PS SL3IncAndTrans(int DeviceAddress,unsigned char CID,unsigned short BNr,unsigned short BNr2,bool MACR,unsigned long Value)</u>	121
<u>2.12.26</u>	<u>MF PS SL3DecAndTrans(int DeviceAddress,unsigned char CID,unsigned short BNr,unsigned short BNr2,bool MACR,unsigned long Value)</u>	122