



mPay Series VC SDK User Manual

Ver: 1.1

Singular Technology

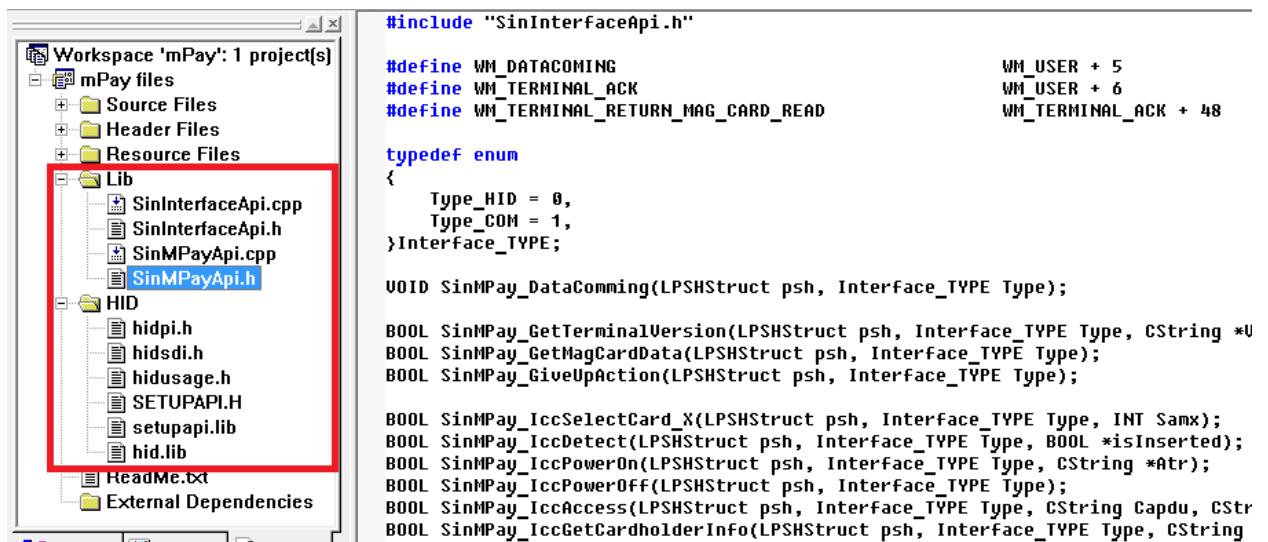
2017/3/14



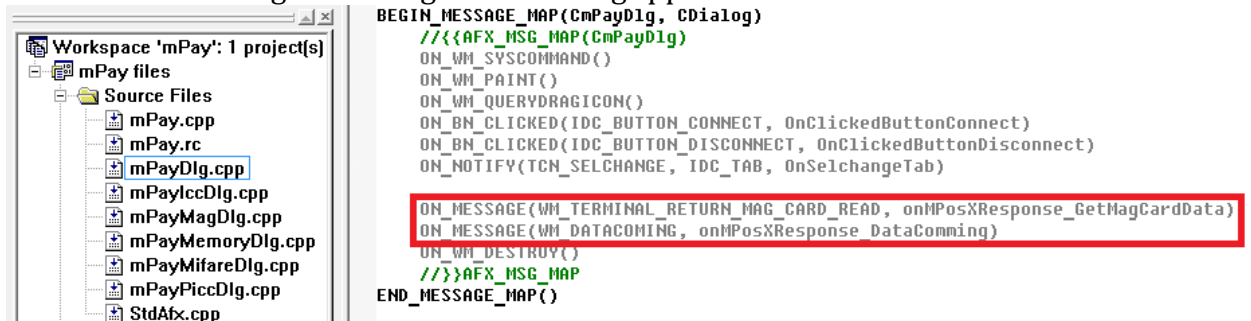
mPay VC SDK User Manual

To use SDK of VC6 demo project

1. First, create an MFC project.
2. Include the source files of mPos Lib and Windows HID
 - a. mPos Lib: Include SinInterfaceApi.h/ SinInterfaceApi.cpp/SinMPayApi.h/ SinMPayApi.cpp, into your project as the demo project.
 - b. Windows HID: Include hippy.h/ hipsdih/hidusage.h/setupapi.h/setupapi.lib/hid.lib, into your project as the demo project.
 - c. You can see the main API list in SinEmvL2.h



- d. You can see the delegate messages in mPosDlg.cpp

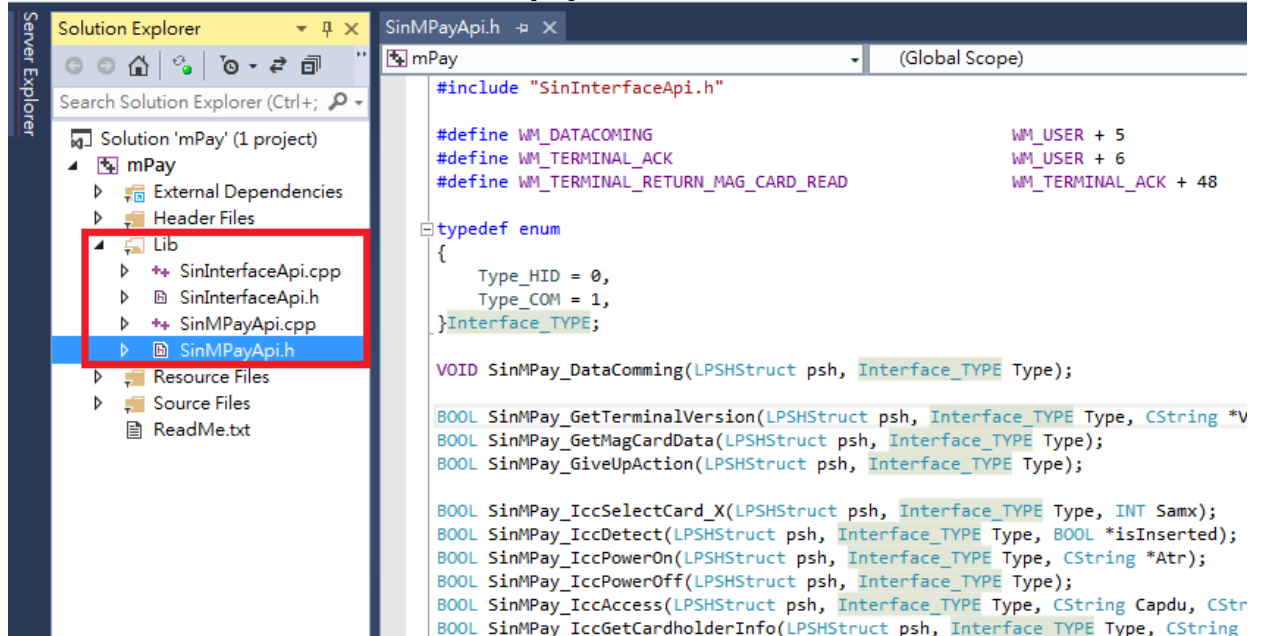


- e. Include <afxmt.h> in StdAfx.h or in any *.cpp of your project for Windows multi-thread and CEvent class API calling.

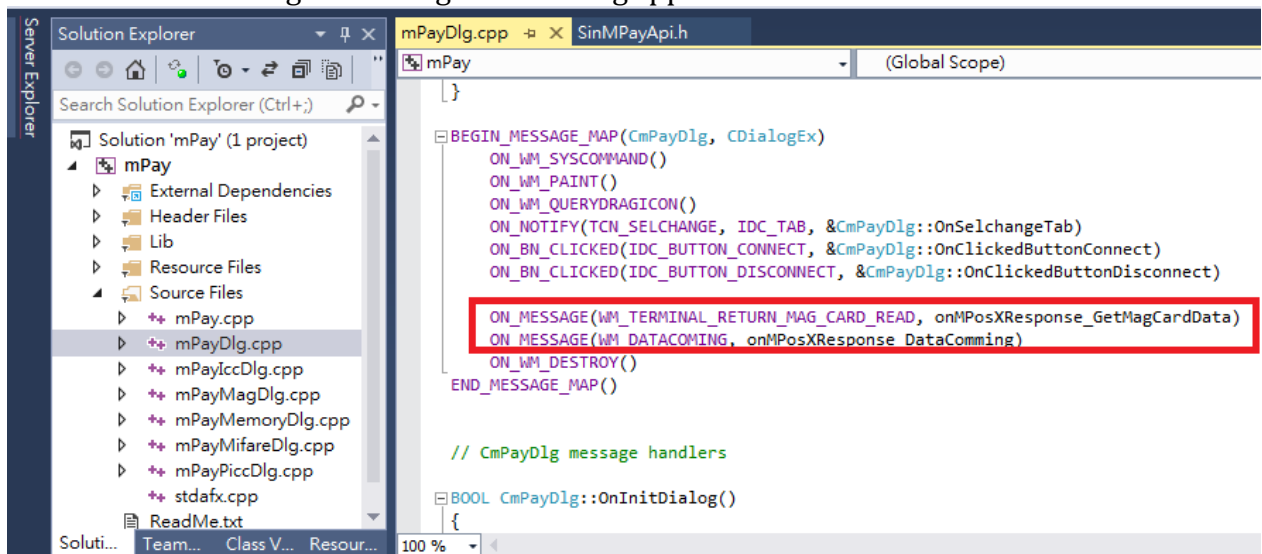


To use SDKof VC2012 demo project and later versions

3. First, create an MFC project.
4. Include the source files of mPay Lib
 - a. mPos Lib: Include SinInterfaceApi.h/ SinInterfaceApi.cpp/SinMPayApi.h/ SinMPayApi.cpp, into your project as the demo project.
 - b. You can see the main API list in SinMPayApi.h



- c. You can see the delegate message in mPosDlg.cpp



To refer to MR5 Command Set

5. You can also refer to the command list under path \MR5_YYYYMMDD\SDK\ MR5_command_set_Vx.x.pdf.



1 SinInterfaceApi.h API List:

S/N	Function Name	Function Description
1	<code>BOOL SinInterface_GetApiVersion(LPSHStruct psh, CString *Version);</code>	Get this API version.
2	<code>VOID GetDeviceCapabilities(LPSHStruct psh);</code>	Get this HID device's capability.
3	<code>BOOL SinHid_GetDevice(LPSHStruct psh, INT VendorID, INT ProductID);</code>	Find a connected HID device which matches Singular's Vendor ID and Product ID and then open this HID port.
4	<code>VOID SinHid_CloseDevice(LPSHStruct psh);</code>	Close this connected HID port.
5	<code>BOOL SinHid_Read(LPSHStruct psh);</code>	Read data through the connected HID port (asynchronous mode).
6	<code>BOOL SinHid_SingleRead(LPSHStruct psh);</code>	Read data through the connected HID port (synchronous mode).
7	<code>UINT SinHid_ReadThread(LPVOID lpParam);</code>	Handle the read thread of the connected HID port (asynchronous mode).
8	<code>BOOL SinHid_Write(LPSHStruct psh, BOOL bAddEDC, BOOL bMaskParameter);</code>	Write data through the connected HID port.
9	<code>BOOL SinHid_SwitchReadHandle(LPSHStruct psh, BOOL fOverlapped);</code>	Switch a device handle in-between synchronous mode or asynchronous mode of reading.
10	<code>BOOL SinComGetandSetTimeouts(LPSHStruct psh, DWORD dwReadInter, DWORD dwReadConstant, DWORD dwReadMultiplier, DWORD dwWriteConstant, DWORD dwWriteMultiplier);</code>	Set timeout parameters of com port communication.
11	<code>BOOL SinCom_Open(LPSCStruct pse, DWORD dwComNum, DWORD dwBaudrate);</code>	Open com port.
12	<code>BOOL SinCom_Close(LPSCStruct pse);</code>	Close com port.
13	<code>BOOL SinCom_Read(LPSHStruct psh);</code>	Read data through the connected com port (asynchronous mode).
14	<code>UINT SinCom_ReadThreadFunction(LPVOID pParam);</code>	Handle the read thread of the connected com port (asynchronous mode).
15	<code>BOOL SinCom_Write(LPSHStruct psh, BOOL bAddEDC, BOOL bMaskParameter);</code>	Write data through the connected com port.



2 SinMPayApi.h API List:

1	<code>VOID SinMPay_DataComming(LPSHStruct psh, Interface_TYPE Type);</code>	Parse read data and send delegate messages.
2	<code>BOOL SinMPay_GetTerminalVersion(LPSHStruct psh, Interface_TYPE Type, CString *Version);</code>	Get terminal version string.
3	<code>BOOL SinMPay_GetMagCardData(LPSHStruct psh, Interface_TYPE Type);</code>	Get magnetic card data.
4	<code>BOOL SinMPay_GiveUpAction(LPSHStruct psh, Interface_TYPE Type);</code>	Give up an ongoing action (command).
5	<code>BOOL SinMPay_IccSelectCard_X(LPSHStruct psh, Interface_TYPE Type, INT Samx);</code>	Select to access the slot for IC card or SAM module.
6	<code>BOOL SinMPay_IccDetect(LPSHStruct psh, Interface_TYPE Type, BOOL *isInserted);</code>	Get IC card slot status (a card inserted or not).
7	<code>BOOL SinMPay_IccPowerOn(LPSHStruct psh, Interface_TYPE Type, CString *Atr);</code>	Power on the inserted IC card and get its ATR.
8	<code>BOOL SinMPay_IccPowerOff(LPSHStruct psh, Interface_TYPE Type);</code>	Power off the inserted IC card.
9	<code>BOOL SinMPay_IccAccess(LPSHStruct psh, Interface_TYPE Type, CString Capdu, CString *Rapdu)</code>	Access the inserted IC card and send a CAPDU command and then get its RAPDU response back.
10	<code>BOOL SinMPay_IccGetCardholderInfo(LPSHStruct psh, Interface_TYPE Type, CString *Pan, CString *Name, CString *Exp);</code>	Get the inserted IC credit card's card holder information.
11	<code>BOOL SinMPay_IccGetTrack2Data(LPSHStruct psh, Interface_TYPE Type, CString *Track2Data);</code>	Get the inserted IC credit card's Track2 information.
12	<code>BOOL SinMPay_MemorySelectCardType(LPSHStruct psh, Interface_TYPE Type, INT CardType);</code>	Select a specified memory card type.
13	<code>BOOL SinMPay_MemoryPowerOn(</code>	Power on the inserted memory card and



	LPSHStruct psh, Interface_TYPE Type, CString *Atr);	get its ATR.
14	BOOL SinMPay_MemoryGetCardType(LPSHStruct psh, Interface_TYPE Type, INT *CardType);	Get the selected memory card type.
15	BOOL SinMPay_MemoryReadData(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString *Data);	Read data from the inserted memory card at specified address and length.
16	BOOL SinMPay_MemoryWriteData(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString Data);	Write data onto the inserted memory card at specified address and length.
17	BOOL SinMPay_MemoryPowerOff(LPSHStruct psh, Interface_TYPE Type);	Power off the inserted memory card
18	BOOL SinMPay_MemoryReadErrorCounter(LPSHStruct psh, Interface_TYPE Type, CString *ErCount);	Read the error counter from the inserted memory card.
19	BOOL SinMPay_MemoryVerifyPsc(LPSHStruct psh, Interface_TYPE Type, CString Length, CString Psc);	Verify the PSC (Programmable Security Code) from the inserted memory card.
20	BOOL SinMPay_MemoryGetPsc(LPSHStruct psh, Interface_TYPE Type, CString Length, CString *Psc);	Read the PSC (Programmable Security Code) from the inserted memory card.
21	BOOL SinMPay_MemoryModifyPsc(LPSHStruct psh, Interface_TYPE Type, CString Length, CString Psc);	Overwrite the PSC (Programmable Security Code) onto the inserted memory card.
22	BOOL SinMPay_MemoryReadDataWithProtectBit(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString *DataWithPb);	Read data with their protect bits from the inserted memory card at specified address and length.
23	BOOL SinMPay_MemoryWriteDataWithProtectBit(LPSHStruct psh, Interface_TYPE Type,	Write data with their protect bits onto the inserted memory card at specified address and length.



	CString Address, CString Length, CString Data);	
24	BOOL SinMPay_MemoryReadProtectionData(LPSHStruct psh, Interface_TYPE Type, CString *Data);	Read all protect bits from the inserted memory card.
25	BOOL SinMPay_MemoryWriteProtectionData(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString Data);	Write protect bits onto the inserted memory card.
26	BOOL SinMPay_PiccActivate(LPSHStruct psh, Interface_TYPE Type);	Active the PICC card in the operating distance of reader antenna.
27	BOOL SinMPay_PiccDeactivate(LPSHStruct psh, Interface_TYPE Type);	De-active the PICC card in the operating distance of reader antenna.
28	BOOL SinMPay_PiccRats(LPSHStruct psh, Interface_TYPE Type, CString *Rats);	Set a data rate in-between the reader and the PICC card.
29	BOOL SinMPay_PiccAccess(LPSHStruct psh, Interface_TYPE Type, CString Capdu, CString *Rapdu);	Access the inserted PICC card and send a CAPDU command and then get its RAPDU response back.
30	BOOL SinMPay_MifareAuth(LPSHStruct psh, Interface_TYPE Type, INT KeyType, INT Sector, CString Key);	Authenticate the Key value of the Sector of the activated PICC card is successful or failed.
31	BOOL SinMPay_MifareReadBlock(LPSHStruct psh, Interface_TYPE Type, INT Block, CString *Data);	Read Data from the Block of the activated PICC card.
32	BOOL SinMPay_MifareWriteBlock(LPSHStruct psh, Interface_TYPE Type, INT Block, CString Data);	Write data onto the block of the activated PICC card.
33	BOOL SinMPay_MifareIncrement(LPSHStruct psh, Interface_TYPE Type, INT Block, CString Value);	Increment a value onto the block of the activated PICC card.
34	BOOL SinMPay_MifareDecrement(Decrement a value onto the block of the



	<code>LPSHStruct psh, Interface_TYPE Type, INT Block, CString Value);</code>	activated PICC card.
35	<code>BOOL SinMPay_GetReaderSerialNumber(LPSHStruct psh, Interface_TYPE Type, CString *SerialNo);</code>	Read and get reader's serial number.
36	<code>BOOL SinMPay_SetReaderSerialNumber(LPSHStruct psh, Interface_TYPE Type, CString SerialNo);</code>	Write and set serial number into reader.

3 Delegate Function List

S/N	Function Name	Function Description
1	onMPosXResponse_GetMagCardData	The delegated function to get magnetic card information after user swiping a card for message WM_TERMINAL_RETURN_MAG_CARD_READ.

4 SinInterfaceApi.h API Description

4.1 `BOOL SinInterface_GetApiVersion(LPSHStruct psh, CString *Version);`

Get this API version.

Parameter:

psh: a point to the communication structure

Version: respond API version string

(e.g., "Sin HID & COM Series Interface Function V1.0").

Return: TRUE: success; FALSE: failure.

4.2 `VOID GetDeviceCapabilities(LPSHStruct psh);`

Get this HID device's capability.

Parameter:

psh: a point to the communication structure

Return: VOID.

4.3 `BOOL SinHid_GetDevice(LPSHStruct psh, INT VendorID, INT ProductID);`

Find a connected HID device which matches Singular's Vendor ID and Product ID and then open this HID port.



Parameter:

psh: a point to the communication structure

VendorID: Singular Technology's USB device Vendor ID (0x273A).

ProductID: MR5's USB device Product ID (0x0131).

Return: TRUE: success; FALSE: failure.

4.4 VOID SinHid_CloseDevice(LPSHStruct psh);

Close this connected HID port.

Parameter:

psh: a point to the communication structure

Return: TRUE: success; FALSE: failure.

4.5 BOOL SinHid_Read(LPSHStruct psh);

Read data through the connected HID port (asynchronous mode).

Parameter:

psh: a point to the communication structure

Return: TRUE: success; FALSE: failure.

4.6 BOOL SinHid_SingleRead(LPSHStruct psh);

Read data through the connected HID port (synchronous mode).

Parameter:

psh: a point to the communication structure

Return: TRUE: success; FALSE: failure.

4.7 UINT SinHid_ReadThread(LPVOID lpParam);

Handle the read thread (asynchronous mode).

Parameter:

lpParam: a point to the parameter for the read thread.

Return: 0 when device handle is INVALID_HANDLE_VALUE or keep checking if data coming.

4.8 BOOL SinHid_Write(LPSHStruct psh, BOOL bAddEDC, BOOL bMaskParameter);

Write data through the connected HID port.

Parameter:

psh: a point to the communication structure

bAddEDC: Add a EDC after a written command or not (always True).



bMaskParameter: Command mask parameter (always FALSE).

Return: TRUE: success; FALSE: failure.

4.9 **BOOL** SinHid_SwitchReadHandle(**LPSHStruct** psh, **BOOL** fOverlapped);
Switch a device handle in-between synchronous mode or asynchronous mode of reading.

Parameter:

psh: a point to the communication structure

fOverlapped: TRUE (asynchronous mode) or FALSE (synchronous mode).

Return: TRUE: success; FALSE: failure.

4.10 **BOOL** SinComGetandSetTimeouts(**LPSHStruct** psh, **DWORD** dwReadInter, **DWORD** dwReadConstant, **DWORD** dwReadMultiplier, **DWORD** dwWriteConstant, **DWORD** dwWriteMultiplier);
Get and set timeout parameters of com port communication.

Parameter:

psh: a point to the communication structure

dwReadInter: Refer to MSDN's COMMTIMEOUT structure.

dwReadConstant: (as the above)

dwReadMultiplier: (as the above)

dwWriteConstant: (as the above)

dwWriteMultiplier: (as the above)

Return: TRUE: success; FALSE: failure.

4.11 **BOOL** SinCom_Open(**LPSCStruct** pse, **DWORD** dwComNum, **DWORD** dwBaudrate);
Open com port.

Parameter:

psh: a point to the communication structure

dwComNum: com port number.

dwBaudrate: com port baud-rate.

Return: TRUE: success; FALSE: failure.

4.12 **BOOL** SinCom_Close(**LPSCStruct** pse);
Close com port.

Parameter:

psh: a point to the communication structure

Return: TRUE: success; FALSE: failure



4.13 **BOOL** SinCom_Read(**LPSHStruct** psh);
Read data through the connected com port (asynchronous mode).

Parameter:

psh: a point to the communication structure

Return: TRUE: success; FALSE: failure.

4.14 **UINT** SinCom_ReadThreadFunction(**LPVOID** pParam);
Handle the read thread the connected com port (asynchronous mode).

Parameter:

lpParam: a point to the parameter for the read thread.

Return: 0 when device handle is INVALID_HANDLE_VALUE or keep checking if data coming.

4.15 **BOOL** SinCom_Write(**LPSHStruct** psh, **BOOL** bAddEDC, **BOOL** bMaskParameter);
Write data through the connected com port.

Parameter:

psh: a point to the communication structure

bAddEDC: Add a EDC after a written command or not (always True).

bMaskParameter: Command mask parameter (always FALSE).

Return: TRUE: success; FALSE: failure.

5 SinMPayApi.h API Description

5.1 **VOID** SinMPay_DataComming(**LPSHStruct** psh, **Interface_TYPE** Type);
Parse read data and send delegate messages.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure.

Delegate: onMPosXResponse_DataComming()

5.2 **BOOL** SinMPay_GetTerminalVersion(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Version);
Get terminal version string.



Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Version: respond version string
(e.g., "MR5 V01.00.00").

Return: TRUE: success; FALSE: failure.

5.3 **BOOL** SinMPay_GetMagCardData(LPSHStruct psh, Interface_TYPE Type);
Get magnetic card data.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure

Delegate: onMPosXResponse_GetMagCardData()

5.4 **BOOL** SinMPay_GiveUpAction(LPSHStruct psh, Interface_TYPE Type);
Give up an ongoing action (command).

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure

5.5 **BOOL** SinMPay_IccSelectCard_X(LPSHStruct psh, Interface_TYPE Type, INT Samx);
Select to access the slot for IC card or SAM module.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Samx: to access the slot for IC card or SAM module (Samx = 0: SmartCard; Samx = 1 : SAM1)

Return: TRUE: success; FALSE: failure

5.6 **BOOL** SinMPay_IccDetect(LPSHStruct psh, Interface_TYPE Type, **BOOL** *isInserted);
Get IC card slot status (a card inserted or not).

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)



isInserted: TRUE: inserted; FALSE: not inserted

Return: TRUE: success; FALSE: failure

5.7 **BOOL** SinMPay_IccPowerOn(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Atr);
Power on the inserted IC card and get its ATR.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Atr: answer-to-reset.

(e.g., "3BEE00008131804280318066B0840C016E01830090008E")

Return: TRUE: success; FALSE: failure

5.8 **BOOL** SinMPay_IccPowerOff(**LPSHStruct** psh, **Interface_TYPE** Type);
Power off the inserted IC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure

5.9 **BOOL** SinMPay_IccAccess(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Capdu, **CString** *Rapdu);
Access the inserted IC card and send a CAPDU command and then get its RAPDU response back.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Capdu: send command to IC card.

(e.g., "00A4040007A0000000031010")

Rapdu: get response from IC card.

(e.g.,

"6F2F8407A0000000031010A524500B56495341204352454449549F380C9F1A029F7A019F02065F2A025F2D027A688701019000")

Return: TRUE: success; FALSE: failure

5.10 **BOOL** SinMPay_IccGetCardholderInfo(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Pan, **CString** *Name, **CString** *Exp);
Get the inserted IC credit card's card holder information.



Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Pan: the card number

Name: the card holder name

Exp: the card expiration date

Return: TRUE: success; FALSE: failure

Delegate: onMPosXResponse_GetCardInfo

5.11 **BOOL** SinMPay_IccGetTrack2Data(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Track2Data);

Get the inserted IC credit card's Track2 data.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Track2Data: the card's Track2 Data

Return: TRUE: success; FALSE: failure

5.12 **BOOL** SinMPay_MemorySelectCardType(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** CardType);

Select a specified memory card type.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

CardType: the memory card type to be accessed later

Return: TRUE: success; FALSE: failure

5.13 **BOOL** SinMPay_MemoryPowerOn(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Atr);

Read data from the inserted memory card at specified address and length.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Atr: answer-to-reset.

(e.g., "A2131091" for SLE4442 card)

Return: TRUE: success; FALSE: failure



5.14 **BOOL** SinMPay_MemoryGetCardType(LPSHStruct psh, Interface_TYPE Type, INT *CardType);
Get the selected memory card type.

Parameter:

psh: a point to the communication structure
Type: the connected Interface type (HID or VCOM)
CardType: the memory card type which was selected

Return: TRUE: success; FALSE: failure

5.15 **BOOL** SinMPay_MemoryReadData(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString *Data);
Read data from the inserted memory card at specified address and length.

Parameter:

psh: a point to the communication structure
Type: the connected Interface type (HID or VCOM)
Address: the start address to be read
Length: the length of bytes to be read
Data: the read data

Return: TRUE: success; FALSE: failure

5.16 **BOOL** SinMPay_MemoryWriteData(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString Data);
Write data onto the inserted memory card at specified address and length.

Parameter:

psh: a point to the communication structure
Type: the connected Interface type (HID or VCOM)
Address: the start address to be written
Length: the length of bytes to be written
Data: the data to be written

Return: TRUE: success; FALSE: failure

5.17 **BOOL** SinMPay_MemoryPowerOff(LPSHStruct psh, Interface_TYPE Type);
Power off the inserted IC card.

Parameter:

psh: a point to the communication structure
Type: the connected Interface type (HID or VCOM)
Atr: answer-to-reset.



(e.g., "A2131091" for SLE4442 card)

Return: TRUE: success; FALSE: failure

5.18 **BOOL** SinMPay_MemoryReadErrorCounter(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *ErCount);

Read the error counter from the inserted memory card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

ErCount: the value of the error counter (hex format)

Return: TRUE: success; FALSE: failure

5.19 **BOOL** SinMPay_MemoryVerifyPsc(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Length, **CString** Psc);

Verify the PSC (Programmable Security Code) from the inserted memory card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Length: the length of PSC to be verified

Psc: the PSC data to be verified

Return: TRUE: success; FALSE: failure

5.20 **BOOL** SinMPay_MemoryGetPsc(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Length, **CString** *Psc);

Read the PSC (Programmable Security Code) from the inserted memory card (after PSC was verified successfully).

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Length: the length of PSC to be read

Psc: the bytes of Psc to be read

Return: TRUE: success; FALSE: failure

5.21 **BOOL** SinMPay_MemoryModifyPsc(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Length, **CString** Psc);

Overwrite the PSC (Programmable Security Code) onto the inserted memory card (after PSC was verified successfully).



Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Length: the length of PSC to be overwritten

Psc: the PSC data to be overwritten

Return: TRUE: success; FALSE: failure

5.22 **BOOL** SinMPay_MemoryReadDataWithProtectBit(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString *DataWithPb);

Read data with their protect bits from the inserted memory card at specified address and length.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Address: the start address to be read

Length: the length of bytes to be read

DataWithPb: the read data with their protect bits

Return: TRUE: success; FALSE: failure

5.23 **BOOL** SinMPay_MemoryWriteDataWithProtectBit(LPSHStruct psh, Interface_TYPE Type, CString Address, CString Length, CString Data);

Write data with their protect bits onto the inserted memory card at specified address and length.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Address: the start address to be written

Length: the length of bytes to be written

Data: the data with their protect bits to be written

Return: TRUE: success; FALSE: failure

5.24 **BOOL** SinMPay_MemoryReadProtectionData(LPSHStruct psh, Interface_TYPE Type, CString *Data);

Read all protect bits from the inserted memory card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Address: the start address to be read

Length: the length of bytes to be read



DataWithPb: the read data with their protect bits

Return: TRUE: success; FALSE: failure

5.25 **BOOL** SinMPay_MemoryWriteProtectionData(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Address, **CString** Length, **CString** Data);
Write protect bits onto the inserted memory card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Address: the start address to be written

Length: the length of bytes to be written

Data: the protect bits to be written

Return: TRUE: success; FALSE: failure

5.26 **BOOL** SinMPay_PiccActivate(**LPSHStruct** psh, **Interface_TYPE** Type);
Active the PICC card in the operating distance of reader antenna.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure

5.27 **BOOL** SinMPay_PiccDeactivate(**LPSHStruct** psh, **Interface_TYPE** Type);
De-active the PICC card in the operating distance of reader antenna.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Return: TRUE: success; FALSE: failure

5.28 **BOOL** SinMPay_PiccRats(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *Rats);
Set a data rate in-between the reader and the PICC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Rats: the data rate in-between the reader and the PICC card.

Return: TRUE: success; FALSE: failure



5.29 **BOOL** SinMPay_PiccAccess(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** Capdu, **CString** *Rapdu);

Access the inserted PICC card and send a CAPDU command and then get its RAPDU response back.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Capdu: send command to IC card.

(e.g., "00A4040007A0000000031010")

Rapdu: get response from IC card.

(e.g.,

"6F2F8407A0000000031010A524500B56495341204352454449549F380C9F1A029F7A019F02065F2A025F2D027A688701019000")

Return: TRUE: success; FALSE: failure

5.30 **BOOL** SinMPay_MifareAuth(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** KeyType, **INT** Sector, **CString** Key);

Authenticate the Key value of the Sector of the activated PICC card is successful or failed.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

KeyType: Key A or Key B

Sector: Select a sector to operate (Sector 0~15).

Key: the key to be authenticated

Return: TRUE: success; FALSE: failure

5.31 **BOOL** SinMPay_MifareReadBlock(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** Block, **CString** *Data);

Read Data from the Block of the activated PICC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Block: Select a Block to operate (Block 0~3).

Data: the read data

Return: TRUE: success; FALSE: failure

5.32 **BOOL** SinMPay_MifareWriteBlock(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** Block, **CString** Data);



Write data onto the block of the activated PICC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Block: Select a Block to operate (Block 0~3).

Data: the data to be written

Return: TRUE: success; FALSE: failure

5.33 **BOOL** SinMPay_MifareIncrement(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** Block, **CString** Value);

Increment a value onto the block of the activated PICC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Block: Select a Block to operate (Block 0~3).

Value: the value to be incremented

Return: TRUE: success; FALSE: failure

5.34 **BOOL** SinMPay_MifareDecrement(**LPSHStruct** psh, **Interface_TYPE** Type, **INT** Block, **CString** Value);

Decrement a value onto the block of the activated PICC card.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

Block: Select a Block to operate (Block 0~3).

Value: the value to be decremented

Return: TRUE: success; FALSE: failure

5.35 **BOOL** SinMPay_GetReaderSerialNumber(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** *SerialNo);

Read and get reader's serial number.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

SerialNo: a point to the string of reader's serial number

Return: TRUE: success; FALSE: failure



5.36 **BOOL** SinMPay_SetReaderSerialNumber(**LPSHStruct** psh, **Interface_TYPE** Type, **CString** SerialNo);

Write and set serial number into reader.

Parameter:

psh: a point to the communication structure

Type: the connected Interface type (HID or VCOM)

SerialNo: a serial number to be written into reader's memory

Return: TRUE: success; FALSE: failure

6 Delegate Function Description

6.1 onMPosXResponse_GetMagCardData

The delegated function to get magnetic card information after user swiping a card for message WM_TERMINAL_RETURN_MAG_CARD_READ.

Revision History:

Revision	Description	Data
V1.0	Initial Release	2017/2/23