

UNION COMMUNITY

Server SDK

Additional functions For Windows

Version 3.0

June, 2009

Software Development Department

UNION COMMUNITY CO., LTd.

Copyright © 2008, UNION COMMUNITY Co., Ltd.
All rights reserved.

USER License Agreement for Software Developer's Kit Designed by Union Community Co., Ltd

This agreement is a legal usage license agreement between Union Community Co., Ltd. and the user.

If you do not agree with the terms and condition of the agreement, please return the product promptly. If you return the product, you will receive a refund.

1. Usage License

UNION COMMUNITY Co., Ltd. Grants licensee to use this SDK a personal, Limited, non-transferable, non-exclusive right to install and use one copy of the SDK on a single computer exclusively.

The software is considered 'being used' if it is stored in a computer's main or other storage device. The number of software copies will be determined by taking the greater number of the number of computers 'used' by the software and the number of computers with the software stored. Licensee may use the SDK solely for developing, designing, and testing UNION software applications for use with UNION products ("Applications").

2. Right to Upgrade

If you have purchased the software by upgrading an older version, the usage license of the old version is transferred to the new version. However, you may only use the old version under the condition that the old and new versions are not running simultaneously. Therefore, you are prohibited from transferring, renting or selling the old version. You maintain the usage license for the program and ancillary files that are in the old version but not in the new version.

3. Assignment of License

If you wish to transfer the usage license of this software to a third party, you must first obtain a written statement indicating that the recipient agrees with this agreement. You must then transfer the original disk and all other program components, and all copies of the program must be destroyed. After the transfer is complete, you must notify UNION COMMUNITY Co., Ltd. to update the customer registration.

Licensee shall not rent, lease, sell or lend the software application developed using the SDK to a third party without UNION's prior written consent.

Licensee shall not copy and redistribute the SDK without UNION's prior written consent. No other uses and/or distribution of the SDK or Sample Code are permitted without UNION's prior written consent. UNION reserves all rights not expressly granted to Licensee.

4. Copyright

All copyrights and intellectual properties of the software and its components belong to UNION COMMUNITY Co., Ltd. and these rights are protected under Korean and international copyright laws. Therefore, you may not make copies of the software other than for your backup purposes. In addition, you may not modify the software other than for reverse-engineering purposes to secure compatibility. Finally, you may not modify, transform or copy any part of the documentation without written permission from UNION COMMUNITY. (If you're using a network product, you may copy the documentation in the amount of the number of users)

5. Installation

An individual user can install this software in his/her PCs at home and office, as well as in a mobile PC. However, the software must not be running from two computers simultaneously. A single product can be installed in two or more computers in one location, but one of those computers must have a usage rate of at least 70%. If another computer has a usage rate of 31% or higher, another copy of the software must be purchased.

6. Limitation of Warranty

UNION COMMUNITY Co., Ltd. guarantees that the CD-ROM and all components are free of physical damage for a year after purchase.

UNION DISCLAIMS ALL WARRANTIES NOT EXPRESSLY PROVIDED IN THIS AGREEMENT INCLUDING, WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE. If you find any manufacture defect within the warranty period, we will replace the product. You must be able to prove that the product has been purchased within a year to receive a replacement, but we will not replace a product damaged due to your mishandling or negligence. UNION COMMUNITY Co., Ltd. does not guarantee that the software and its features will satisfy your specific needs, and is not liable for any consequential damages arising out of the use of this product.

7. Liabilities

UNION COMMUNITY Co., Ltd. is not liable for any verbal, written or other agreements made by third parties, including product suppliers and dealers.

8. Termination

This agreement is valid until the date of termination. However, the agreement shall terminate automatically if you damage the program or its components, or fail to comply with the terms described in this agreement.

9. Customer Service

UNION COMMUNITY Co., Ltd. makes every effort to provide registered customers with technical assistance and solutions to problems regarding software applications under certain system environments. When a customer submits a suggestion about any inconvenience or anomaly experienced during product usage, UNION COMMUNITY Co., Ltd. will take corrective action and notify the customer of the result.

10. General Terms

You acknowledge that you have read, understood and agree with the terms of this agreement. You also recognize the fact that this agreement has precedence over user agreements of older versions, past order agreements, advertisement notifications and/or other written agreements.

11. Contact

If you have any questions about this agreement, please contact UNION COMMUNITY Co., Ltd. via telephone, fax or e-mail.

CONTENTS

| | |
|-------------------------------------|----|
| 1. Glossary of terms | 7 |
| 2. UCSAPI types and macros..... | 8 |
| UCSAPI_PERIPHERAL_DEVICE..... | 8 |
| 3. UCSAPI functions | 9 |
| UCSAPI_ControlPeripheralDevice..... | 9 |
| UCSAPI_EnrollFromTerminal..... | 10 |
| UCSAPI_FreeExportData..... | 11 |
| 4. UCSCOM method and property | 13 |
| 4.1 Properties..... | 13 |
| LONG ErrorCode | 13 |
| LONG TotalFingerCount | 13 |
| LONG FingerID | 14 |
| LONG SampleNumber..... | 14 |
| LONG FPSampleData | 14 |
| LONG FPSampleDataLength..... | 15 |
| 4.2 Methods | 17 |
| ControlPeripheralDevice | 17 |
| EnrollFromTerminal | 18 |

1. Glossary of terms

2. UCSAPI types and macros

UCSAPI_PERIPHERAL_DEVICE

It includes the info to control peripheral devices that are connected to the terminal.

The detail info such as ID, control order and error code on peripheral devices will be defined when devices are connected to the terminal

```
typedef struct ucsapi_peripheral_device
{
    int32_t DeviceID;
    int32_t ControlCommand;
} UCSAPI_PERIPHERAL_DEVICE;
```

DeviceID: Peripheral device ID which is conneted to the terminal.

ControlCommand: Command for controlling peripheral devices.

3. UCSAPI functions

UCSAPI_ControlPeripheralDevice

```
UCSAPI_RETURN UCSAPI UCSAPI_ControlPeripheralDevice(  
    IN int32_t ClientID,  
    IN int32_t TerminalID,  
    IN UCSAPI_PERIPHERAL_DEVICE* PeripheralDevice)
```

Version

Supporting from over UCSAPI.dll Version 1.0.1.2

Definition

This function is to control peripheral devices which are connected to the terminal

Parameter

ClientID : Client ID

TerminalID : Terminal ID

PeripheralDevice : Structure pointer which contains the info on peripheral devices

Return

UCSAPIERR_NONE value is success. All other values mean errors.

Error

UCSAPIERR_NOT_ACTIVE

UCSAPIERR_INVALID_POINTER

UCSAPIERR_INVALID_TERMINAL

Callback Event

UCSAPI_EVENT_CONTROL_PERIPHERALDEVICE

Callback Parameter

wParam : UCSAPI_PACKET_INFO

lParam : UCSAPI_PERIPHERAL_DEVICE* PeripheralDevice

UCSAPI_EnrollFromTerminal

```
UCSAPI_RETURN UCSAPI UCSAPI_EnrollFromTerminal(  
    IN int32_t ClientID,  
    IN int32_t TerminalID,  
    OUT UCBioAPI_EXPORT_DATA_PTR pFingerData)
```

Version

Supporting over UCSAPI.dll version 1.0.4.1

Definition

This function makes terminal to work as a USB fingerprint reader when there is no USB fingerprint reader connected. It makes fingerprint template with one person's fingerprint getting from terminal. pFingerData which has been acquired by this process has to be canceled using UCSAPI_FreeExportData function after using it. The terminal has to be connected to the server since this function uses terminal.

Parameter

ClientID : Client ID

TerminalID : Terminal ID

pFingerData : Structure pointer including Fingerprint Template data. Refer to UCBioBSP SDK UCBioAPI_EXPORT_DATA structure.

Return

UCSAPIERR_NONE value is success. All other values mean errors.

Error

UCSAPIERR_NOT_ACTIVE
UCSAPIERR_INVALID_POINTER
UCSAPIERR_INVALID_TERMINAL

Callback Event

None

Callback Parameter

None

UCSAPI_FreeExportData

UCSAPI_RETURN UCSAPI UCSAPI_FreeExportData(
 IN UCBioAPI_EXPORT_DATA_PTR pFingerData)

Version

Supporting from over UCSAPI.dll version 1.0.3.2

Definition

This function cancels the memory of UCBioAPI_EXPORT_DATA structure made by UCSAPI_EnrollFromTerminal function.

Parameter

pFingerData : UCBioAPI_EXPORT_DATA structure pointer to be canceled. If this value is NULL, there is nothing else to do. Refer to UCBioBSP SDK의 UCBioAPI_EXPORT_DATA structure.

Return

UCSAPIERR_NONE value is success. All other values mean errors.

Error

None

Callback Event

None

Callback Parameter

None

4. UCSCOM method and property

4.1 Properties

Various properties of UCSCOMObj Object are described.

LONG ErrorCode

Prototype:

[ReadOnly] long ErrorCode;

Description:

Values for errors occurred during the setting of executed methods and properties are stored.

The value of 0 means success and all other values mean failure.

Errors occurred during the setting of the method and property of a child object can also be obtained using this value

LONG TotalFingerCount

Prototype:

[ReadOnly] long TotalFingerCount;

Description:

The total number of fingers of converted FIR is stored.

It is used only after the EnrollFromTerminal method is called.

Related methods:

EnrollFromTerminal

Related properties:

FingerID

LONG FingerID

Prototype:

[ReadOnly] long FingerID(long nIndex);

Description:

The finger ID information of converted FIR is stored as array.
nIndex can have a value ranging in 0 ~ (TotalFingerCount – 1).
It is used only after the EnrollFromTerminal method is called.

Related methods:

EnrollFromTerminal

Related properties:

FPSampleDataLength, FPSampleData

LONG SampleNumber

Prototype:

[ReadOnly] long SampleNumber;

Description:

The number of finger-by-finger templates of converted FIR is stored. The value of either 1 or 2 is stored. It is used only after the EnrollFromTerminal method is called.

Related methods:

EnrollFromTerminal

Related properties:

FPSampleDataLength, FPSampleData

LONG FPSampleData

Prototype:

[ReadOnly] VARIANT FPSampleData(
 long nFingerID,
 long SampleNum);

Description:

Binary stream data of finger-by-finger templates of converted FIR are obtained.
nFingerID and SampleNum can be obtained using FingerID and SampleNumber property.

It is used only after the EnrollFromTerminal method is called.

Parameters:

nFingerID:

The ID of a finger to be obtained.

nSampleNumber:

The number of a sample to be obtained. The value of either 0 or 1 is used.

Related methods:

EnrollFromTerminal

Related properties:

FingerID, SampleNumber, FPSampleDataLength

LONG FPSampleDataLength

Prototype:

[ReadOnly] long FPSampleDataLength(
long nFingerID,
long SampleNum);

Description:

The data size of finger-by-finger template of converted FIR is obtained.
nFingerID and SampleNum can be obtained using FingerID 및 SampleNumber property.
It is used only after the EnrollFromTerminal method is called.

Parameters:

nFingerID:

The ID of a finger to be obtained.

nSampleNumber:

The number of a sample to be obtained. The value of either 0 or 1 is used.

Related methods:

EnrollFromTerminal

Related properties:

FingerID, SampleNumber, FPSampleData

4.2 Methods

ControlPeripheralDevice

(LONG Handle,
LONG TerminalID,
LONG PeripheralDeviceID,
LONG ControlCommand)

Version

Supporting from over UCSCOM.dll version 1.0.1.2

Definition

This method initializes UCSCOM module and operates as the server.

Parameter

Handle : Client ID.

TerminalID : Terminal ID.

PeripheralDeviceID : Peripheral devices ID connected to the terminal.

ControlCommand : Command to control peripheral devices.

Return

UCSAPIERR_NONE value is success. All other values mean errors.

Error

UCSAPIERR_NOT_ACTIVE

UCSAPIERR_INVALID_POINTER

UCSAPIERR_INVALID_TERMINAL

Related Properties

ErrorCode

Related Events

EventControlPeripheralDevice(LONG Handle, LONG TerminalID, LONG PeripheralDeviceID, LONG ControlCommand)

Event Parameter

Handle : Client ID

TerminalID: Terminal ID

PeripheralDeviceID : Peripheral devices ID connected to the terminal..

ControlCommand : Command to control peripheral devices.

EnrollFromTerminal

(LONG ClientID,
LONG TerminalID)

Version

Supporting from over UCSCOM.dll version 1.0.3.2

Definition

This method initializes UCSCOM module and operates as the server.

This method makes terminal to work as a USB fingerprint reader when there is no USB fingerprint reader connected. It makes fingerprint template with one person's fingerprint getting from terminal. The terminal has to be connected to the server since this methods uses terminal.

Parameter

Handle : Client ID.

TerminalID : Terminal ID.

Return

UCSAPIERR_NONE value is success. All other values mean errors.

Error

UCSAPIERR_NOT_ACTIVE

UCSAPIERR_INVALID_POINTER

UCSAPIERR_INVALID_TERMINAL

Related Properties

ErrorCode

FPSampleData
TotalFingerCount
FingerID
FPSampleDataLength
SampleNumber

Related Event

None

Event Parameter

ClientID : Client ID
TerminalID: Terminal ID