

SDK/J Authentication Package v1.0 Sample Application

SDK/J Authentication Package Version:1.0



Copyright © 2011 Ricoh Co., Ltd.

Terms of Use and Trademarks

1. The contents of this book may be changed without notice in the future.
2. The copying, reproducing, changing, quoting, reprinting, or distributing a part/all of this book are prohibited.
3. We make no warranty, express or implied, regarding this document and the sample codes described in this document. We will not be held responsible for any of our customer's losses, damages resulting from lost profits, or claims from any third party on using this document and the sample codes described in this document.

4. Trademarks

PostScript® and Acrobat® are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Java, JVM (CVM) and CDC are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Eclipse is a trademark of eclipse.org in the United States, other countries, or both.

OSGi(TM) is a trademark, registered trademark, or service mark of The Open Services Gateway Initiative in the US and other countries.

Apache is a registered trademark of The Apache Software Foundation in the United States, other countries, or both.

Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights in those marks.

Contents

1. Introduction	3
1.1. Target Readers.....	3
2. SCF CardObject Sample	4
2.1. Execution Environment.....	4
2.2. Features Overview	4
2.3. Screen Image	5
2.4. Operation Procedures	6
3. SCF waitForCard Sample.....	7
3.1. Execution Environment.....	7
3.2. Features Overview	7
3.3. Screen Image	8
3.4. Operation Procedures	8
4. SCF CardEvent Sample.....	9
4.1. Execution Environment.....	9
4.2. Features Overview	9
4.3. Screen Image	10
4.4. Operation Procedures	10
5. SCF CardService Sample.....	12
5.1. Execution Environment.....	12
5.2. Features Overview	12
6. OCF waitForCard Sample	13
6.1. Execution Environment.....	13
6.2. Features Overview	13
6.3. Screen Image	14
6.4. Operation Procedures	14
7. OCF CardTerminalEvent Sample	15
7.1. Execution Environment.....	15
7.2. Features Overview	15
7.3. Screen Image	16
7.4. Operation Procedures	16
8. OCF CardService Sample	18
8.1. Execution Environment.....	18

8.2. Features Overview	18
9. Panel Service Sample.....	19
9.1. Operation Environment.....	19
9.2. Features Overview	19
9.3. Screen Images	20
9.4. Usage	21
10. Smart card authentication sample.....	22
10.1. Operation Environment.....	22
10.2. Features Overview	22
10.3. Screen Images	23
10.4. Usage	27
11. Proximity card authentication sample.....	28
11.1. Operation Environment.....	28
11.2. Features Overview	28
11.3. Screen Images	29
11.4. Usage	33
11.5. Card Reader Configuration	34
11.5.1. Configuring the working setting for the card reader	35
11.5.2. Checking the operation of the card reader.....	40
11.5.3. Using another card reader than the supported card readers	41
Change History	42

1. Introduction

SDK/J Authentication Package offers some sample applications.

This document provides information on the sample applications and explains how to use them.

1.1. Target Readers

This document is intended for the Device SDK Type-J (hereinafter referred to as "the SDK/J") application developers who are familiar with:

- The basics of SDK/J application development.
- The basics of the SDK/J Authentication Package.

For details on SDK/J application development, see the Device SDK Type-J Developer's Guide.

For details on SDK/J Authentication Package, see each User's Guide and Developer's Guide.

2. SCF CardObject Sample

Location : smartcard framework/sample/dsdk/dist/285409953

2.1. Execution Environment

- PC/SC daemon is enabled. *

* Please see "SDK/J Authentication Package Settings Guide" for details.

2.2. Features Overview

- The SCF CardObject Sample is an Xlet application that retrieves information from the card, sends an APDU request to the card, and receives an APDU response from the card, by using an instance of the Card class of the SCF.
- This sample application supports all sorts of cards the card reader device can read; the range of the supported cards is determined depending on the specification of the card reader device.
- When an error has occurred, this sample application displays an error message on the operation panel.

2.3. Screen Image

SmartCard Framework CardObject sample

(1) Set card.

Slot : (2)

Protocol : (3)

ATR : (4)

(5) Enter Request : (6)

Response : (7)

System Status Job List 2006/ 9/26 16:51

(1) Message Display Area

A message regarding the operation of the application is displayed.
It can be an instruction, an error message, etc. for example.

(2) Slot Display Area

The slot to which the card has been inserted is displayed.

(3) Protocol Display Area

The protocol the card uses is displayed.

(4) ATR Display Area

The ATR of the card is displayed.

(5) Enter Button

When this button is clicked, a soft keyboard that allows the user to specify the APDU request to send to the card appears. The APDU request should be specified by a hex string.

(6) Request Display Area

The APDU request to be sent to the card is displayed.

(7) Response Display Area

The response APDU the card returns is displayed.

2.4. Operation Procedures

- Card Detection
 - Set a card to the card reader device.
 - When the APDU request has been specified, the application will send the request to the card and receive the APDU response from the card.
 - When the card is detected, the [Start] key LED indicator will glow green.
- Card Removal
 - Remove the card from the card reader device.
 - When the card is removed from the card reader device, the [Start] key LED indicator will glow red.

3. SCF waitForCard Sample

Location : smartcard framework/sample/dsdk/dist/285409954

3.1. Execution Environment

- PC/SC daemon is enabled. *
- SCF CardService Sample bundle is installed and is operating properly.

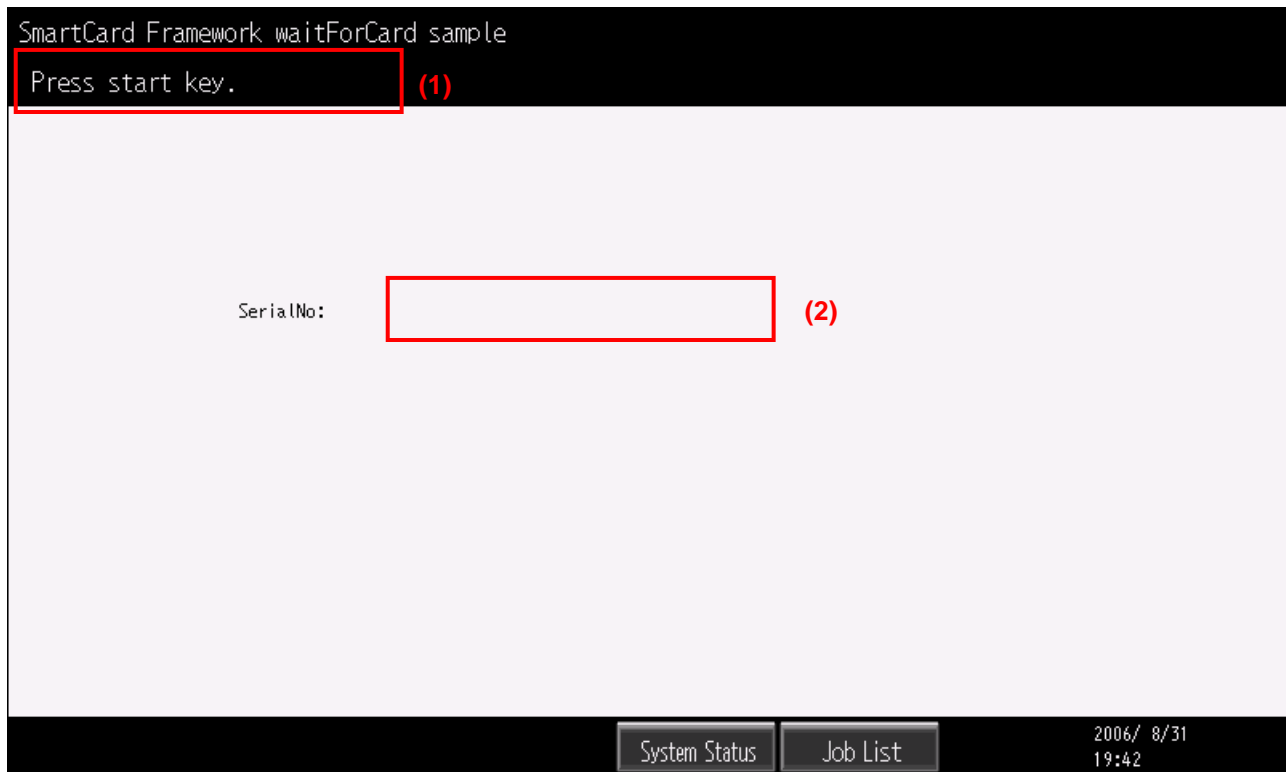
* Please see “SDK/J Authentication Package Settings Guide” for details.

3.2. Features Overview

- The SCF waitForCard Sample is an Xlet application that detects the card by using the waitForCard method of the SCF.
- This sample application obtains the serial number of the Cryptoflex card by using the card service offered by the SCF CardService Sample bundle after detecting the card, and then displays it on the operation panel.
- When an error has occurred, this sample application displays an error message on the operation panel.

* For the supported Cryptoflex cards, see the table on page 12.

3.3. Screen Image



(1) Message Display Area

A message regarding the operation of the application is displayed.
It can be an instruction, an error message, etc. for example.

(2) Serial Number Display Area

The serial number that has been obtained from the card is displayed.

3.4. Operation Procedures

- Start of the waitForCard method
 - Press the [Start] key.
 - The waitForCard method will start and the [Start] key LED indicator will blink red.
- Card Detection → Serial Number Obtainment
 - Set a card to the card reader device.
 - After the serial number is obtained, the waitForCard method will stop and the [Start] key LED indicator will glow green.
- Stop of the waitForCard method
 - Press the [Clear/Stop] key.
 - The waitForCard method will stop and the [Start] key LED indicator will glow green.

4. SCF CardEvent Sample

Location : smartcard framework/sample/dsdk/dist/285409955

4.1. Execution Environment

- PC/SC daemon is enabled. *
- SCF CardService Sample bundle is installed and is operating properly.

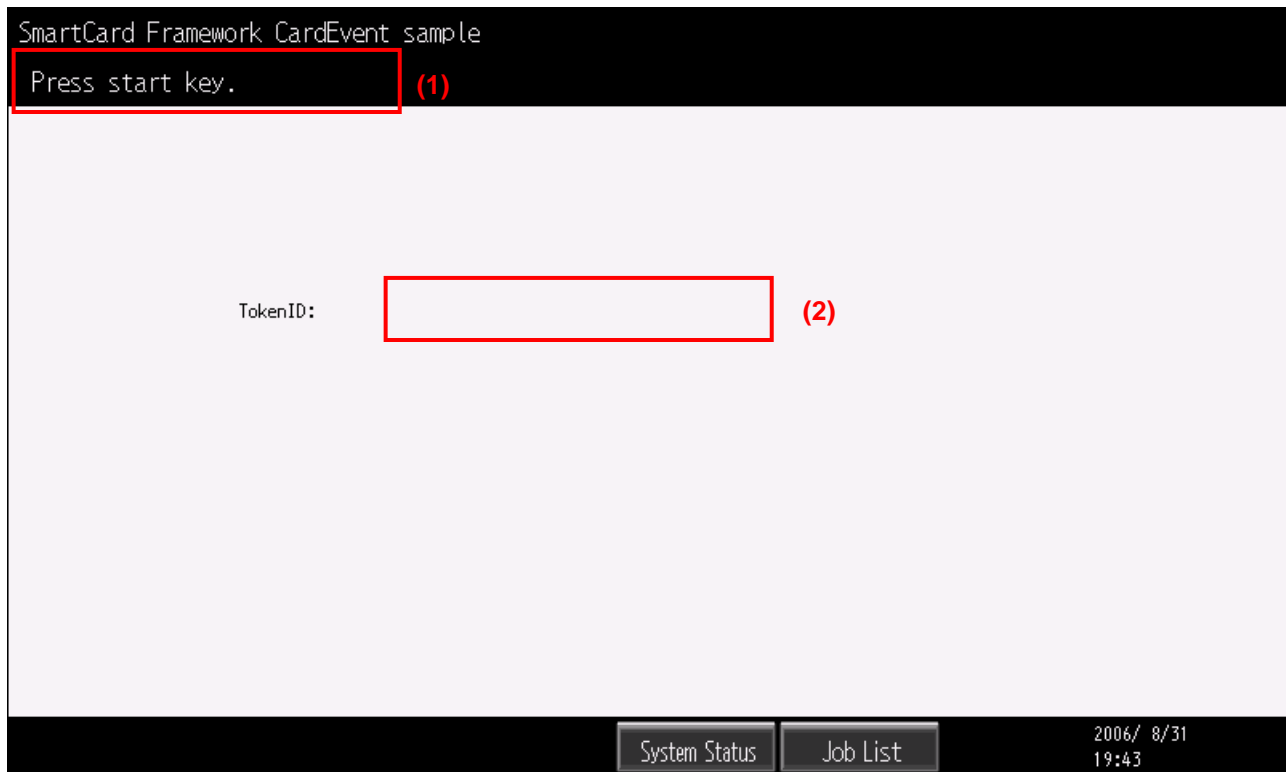
* Please see “SDK/J Authentication Package Settings Guide” for details.

4.2. Features Overview

- The SCF CardEvent Sample is an Xlet application that detects the card by using the CardEvent class of the SCF.
- This sample application obtains the token ID of the JavaCard card by using the card service offered by the SCF CardService bundle, after detecting the card, and then displays it on the operation panel.
- Also, when the card is removed from the card reader device, this sample application displays a message that indicates the card removal on the operation panel.
- When an error has occurred, this sample application displays an error message on the operation panel.

* For the supported JavaCard cards, see the table on page 12.

4.3. Screen Image



(1) Message Display Area

A message regarding the operation of the application is displayed.
It can be an instruction, an error message, etc. for example.

(2) Token ID Display Area

The token ID that has been obtained from the card is displayed.

4.4. Operation Procedures

- Registration of the CardEvent listener
 - Press the [Start] key.
 - After the CardEvent listener is registered, the CardEvent listener will start and the [Start] key LED indicator will blink red.
- Card Detection → Token ID Obtainment
 - Set a card to the card reader device.
 - After the token ID is obtained, the [Start] key LED indicator will blink green.
- Card Removal Detection
 - Remove the card from the card reader device.
 - After the card removal is detected, the [Start] key will blink red.

- Deletion of the CardEvent listener
 - Press the [Clear/Stop] key.
 - The CardEvent listener will be unregistered and the [Start] key LED indicator will glow green.

5. SCF CardService Sample

Location : smartcard framework/sample/server/dist/285409956

5.1. Execution Environment

- PC/SC daemon is enabled. *

* Please see “SDK/J Authentication Package Settings Guide” for details.

5.2. Features Overview

The SCF CardService Sample is the OSGi bundle that registers/unregisters the scfbundle.MyAppletService service and the scfbundle.MyFileAccessService service with/from the card service registry.

When the bundle starts, the MyAppletService service and the MyFileAccessService service are registered with the card service registry. And when the bundle stops, they are unregistered from the card service registry.

This means that they will be available as long as the bundle is operating.

The MyAppletService service is a card service that, in a JavaCard card, selects an applet and sends an APDU request to the applet.

The MyFileAccessService service is a card service that, in a Cryptoflex card, reads files.

The supported cards are as follows:

MyAppletService	MyFileAccessService
The JavaCard cards that comply with either of the following standards:	Cryptoflex 4k , Cryptoflex 8k ,
- JavaCard2.1.X + GlobalPlatform2.0.X	Cryptoflex 8k v2 , Cryptoflex 16k ,
- JavaCard2.2.X + GlobalPlatform2.1.X	Cryptoflex 32k , Cyptoflex 32k v1 ,
* The CardManager of the GlobalPlatform must be specified as the default applet.	Cryptoflex e-gate, Cryptoflex e-gate 32k

6. OCF waitForCard Sample

Location : OpenCard Framework/sample/dsdk/dist/285409957

6.1. Execution Environment

- PC/SC daemon is enabled. *
- OCF CardService Sample bundle is installed and is operating properly.

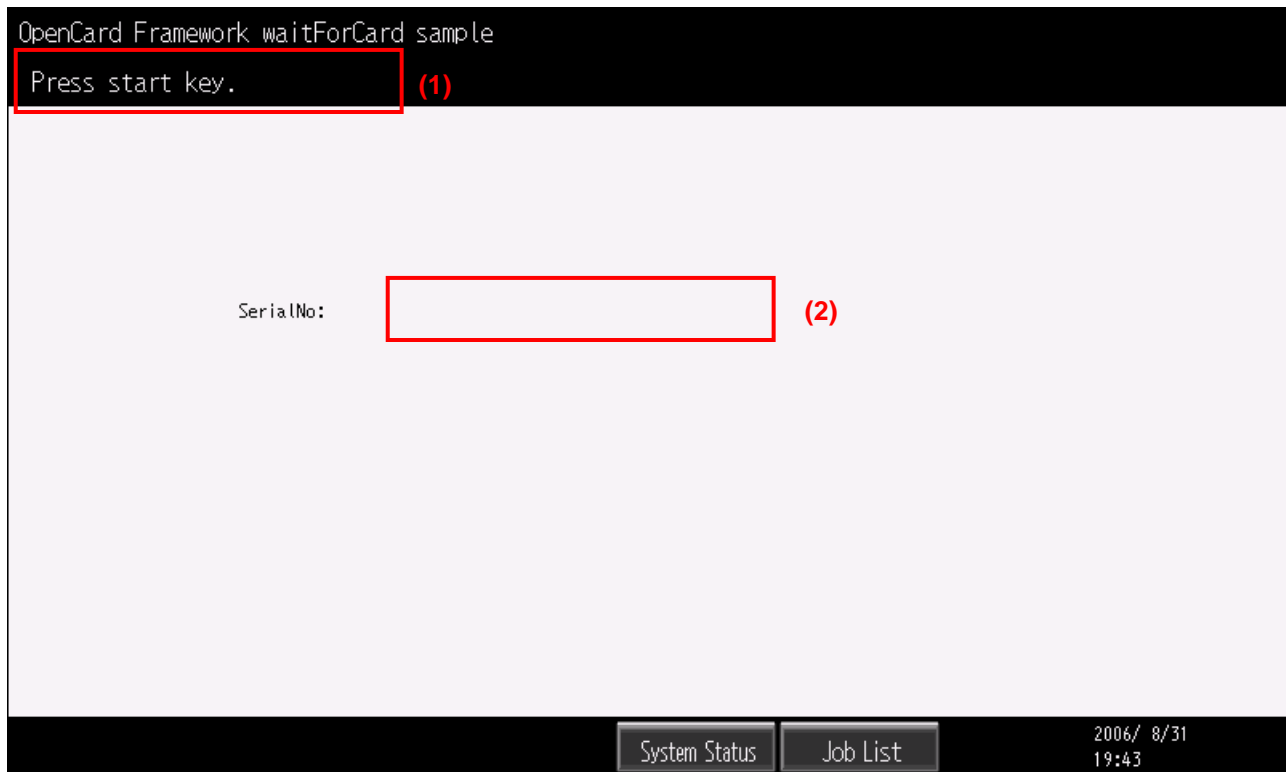
* Please see “SDK/J Authentication Package Settings Guide” for details.

6.2. Features Overview

- The OCF waitForCard Sample is an Xlet application that detects the card by using the waitForCard method of the OCF.
- This sample application obtains the serial number of the Cryptoflex^{*} card by using the card service offered by the OCF CardService Sample bundle after detecting the card, and then displays the obtained serial number on the operation panel.
- When an error has occurred, this sample application displays an error message on the operation panel.

* For the supported JavaCard cards, see the table on page 18.

6.3. Screen Image



(1) Message Display Area

A message regarding the operation of the application is displayed.
It can be an instruction, an error message, etc. for example.

(2) Serial Number Display Area

The serial number that has been obtained from the card is displayed.

6.4. Operation Procedures

- Start of the waitForCard method
 - Press the [Start] key.
 - The waitForCard method will start and the [Start] key LED indicator will blink red.
- Card Detection --> Serial Number Obtainment
 - Set a card to the card reader device.
 - After the serial number is obtained, the waitForCard method will stop and the [Start] key LED indicator will glow green.
- Stop of the waitForCard method
 - Press the [Clear/Stop] key.
 - The waitForCard method will stop and the [Start] key will glow green.

7. OCF CardTerminalEvent Sample

Location : opencard framework/sample/dsdc/dist/285409958

7.1. Execution Environment

- PC/SC daemon is enabled. *
- OCF CardService Sample bundle is installed and is operating properly.

* Please see “SDK/J Authentication Package Settings Guide” for details.

7.2. Features Overview

- The OCF CardTerminalEvent Sample is an Xlet application that detects the card by using the CardTerminalEvent class of the OCF.
- This sample application obtains the token ID of the JavaCard card by using the card service offered by the OCF CardService Sample bundle after detecting the card, and then displays the obtained token ID on the operation panel.
- Also, when the card is removed from the card reader device, this sample application displays a message that indicates the card removal on the operation panel.
- When an error has occurred, this sample application displays an error message on the operation panel.

* For the supported JavaCard cards, see the table on page 18.

7.3. Screen Image



(1) Message Display Area

A message regarding the operation of the application is displayed.
It can be an instruction, an error message, etc. for example.

(2) Token ID Display Area

The token ID that has been obtained from the card is displayed.

7.4. Operation Procedures

- Registration of the CardEvent listener
 - Press the [Start] key.
 - After the CardEvent listener is registered, the CardEvent listener will start and the [Start] key LED indicator will blink red.
- Card Detection → Token ID Obtainment
 - Set a card to the card reader device.
 - After the token ID is obtained, the [Start] key LED indicator will blink green.
- Card Removal Detection
 - Remove the card from the card reader device.
 - After the card removal is detected, the [Start] key LED indicator will blink red.

- Deletion of the CardEvent listener
 - Press the [Clear/Stop] key.
 - The CardEvent listener will be unregistered and the [Start] key LED indicator will glow green.

8. OCF CardService Sample

Location : opencard framework/sample/server/dist/285409959

8.1. Execution Environment

- PC/SC daemon is enabled. *

* Please see “SDK/J Authentication Package Settings Guide” for details.

8.2. Features Overview

The OCF CardService Sample is the OSGi bundle that registers/unregisters the ocfbundle.MyFactory service with/from the card service registry.

When the bundle starts, the MyFactory service is registered with the card service registry. And when the bundle stops, it is unregistered from the card service registry.

This means that the card services offered by the MyFactory service, the ocfbundle.MyAppletService service and the ocfbundle.MyFileAccessService service, will be available as long as the bundle is operating.

The MyAppletService service is a card service that, in a JavaCard card, selects an applet and sends an APDU request to the applet.

The MyFileAccessService service is a card service that, in a Cryptoflex card, reads files.

The supported cards are as follows:

MyAppletService	MyFileAccessService
The JavaCard cards that comply with either of the following standards:	Cryptoflex 4k , Cryptoflex 8k ,
- JavaCard2.1.X + GlobalPlatform2.0.X	Cryptoflex 8k v2 , Cryptoflex 16k ,
- JavaCard2.2.X + GlobalPlatform2.1.X	Cryptoflex 32k , Cyptoflex 32k v1 ,
* The CardManager of the GlobalPlatform must be specified as the default applet.	Cryptoflex e-gate, Cryptoflex e-gate 32k

9. Panel Service Sample

Location : panelservice/sample/server/dist/285409960

9.1. Operation Environment

- PC/SC daemon is enabled. *
- Panel Service API is enabled. *
- No other SDK/J application using the Panel Service is currently active on the same device.
- The address book of the target MFP/LP has the user whose login user name and the login password are “user” and “pass” respectively. (For details on the address book configuration, see the document supplied with the target device.)
- The basic user authentication management and the enhanced authentication management are enabled on the target device. (For details on the authentication function configuration, see the Panel Service User's Guide.)

* Please see “SDK/J Authentication Package Settings Guide” for details.

9.2. Features Overview

This sample application is a server type application which provides the following features:

- Card detection: when a card is set to the card reader device, it is detected.
- PIN verification: a PIN verification is performed after a card is detected.
- Obtainment of the login user name and login password: the login user name and login password is obtained after the PIN verification.
- Unlock of the operation panel: the obtained login user name and login password is sent to the user authentication module and the operation panel is unlocked.

This sample application uses SCF to communicate with smart cards. Any smart card type can be used in this sample application as long as it is supported by the PC/SC card reader connected to the target device.

This sample application uses the following parameter values; they can not be changed.

Parameter	Value
PIN	1234
Login user name	user
Login password	pass

9.3. Screen Images

<<Web screen>>

This sample application has no screen.

<<Panel image>>

When the authentication management is set appropriately, the panel image will be as follows. (For 4 line LCD models, the panel image will be different one)

(1) Login User Name

Enter (2)

Login Password

Enter

(3) Cancel

(4) Login

(1) Login User Name Display Area and Login Password Display Area

These areas display the login username and login password entered from the soft keyboard.

(2) Enter Buttons (Soft keyboard display buttons)

These buttons display a soft keyboard when they are clicked.

(3) Cancel Button

When this button is clicked, the entered user name and/or password are canceled.

(4) Login Button

When this button is clicked, a user authentication is performed with the given login user name and login password.

Note: This UI is for manual input authentication.

9.4. Usage

Login

Set a card to the card reader device.

(For manual login, click the Enter button and enter a login user name and login password by using the soft keyboard.)

Logout

Press the [Login/Logout] hard-key when you are in login.

10. Smart card authentication sample

Location : sample/server/dist/285409977

URL : http://[ipaddress]:8080/smart/localauth/main

10.1. Operation Environment

- PC/SC daemon is enabled. *
- Panel Service API is enabled. *
- No other SDK/J application using the Panel Service is currently active on the same device.
- The address book of the target MFP/LP has users whose login user name and the login password are set appropriately. (For details on the address book configuration, see the document supplied with the target device.)
- The basic user authentication management and the enhanced authentication management are enabled on the target device. (For details on the authentication function configuration, see the Panel Service User's Guide.)

* Please see "SDK/J Authentication Package Settings Guide" for details.

10.2. Features Overview

This sample application is a server type application using Panel Service. The work flow of this sample is as follows:

1. Detects a smart card.
2. Obtains theID of the inserted smart card.
3. Converts the ID to the corresponding login user name and login password using the local ID list.
4. Performs the user authentication using the login user name and login password.

If the login user name and login password correspond which correspond to the inserted card ID can not be found in the ID list, the user authentication will fail. The ID list can be modified from web screen or by editing [INSTALLPATH]/home/localauth.dat file directly.

This sample application uses SCF to communicate with smart cards. The smart cards available in this sample application are as follows:

- JavaCard

JavaCard2.1.X + GlobalPlatform2.0.X or JavaCard2.2.X + GlobalPlatform2.1.X

(The CardManager of the GlobalPlatform must be specified as the default applet.)

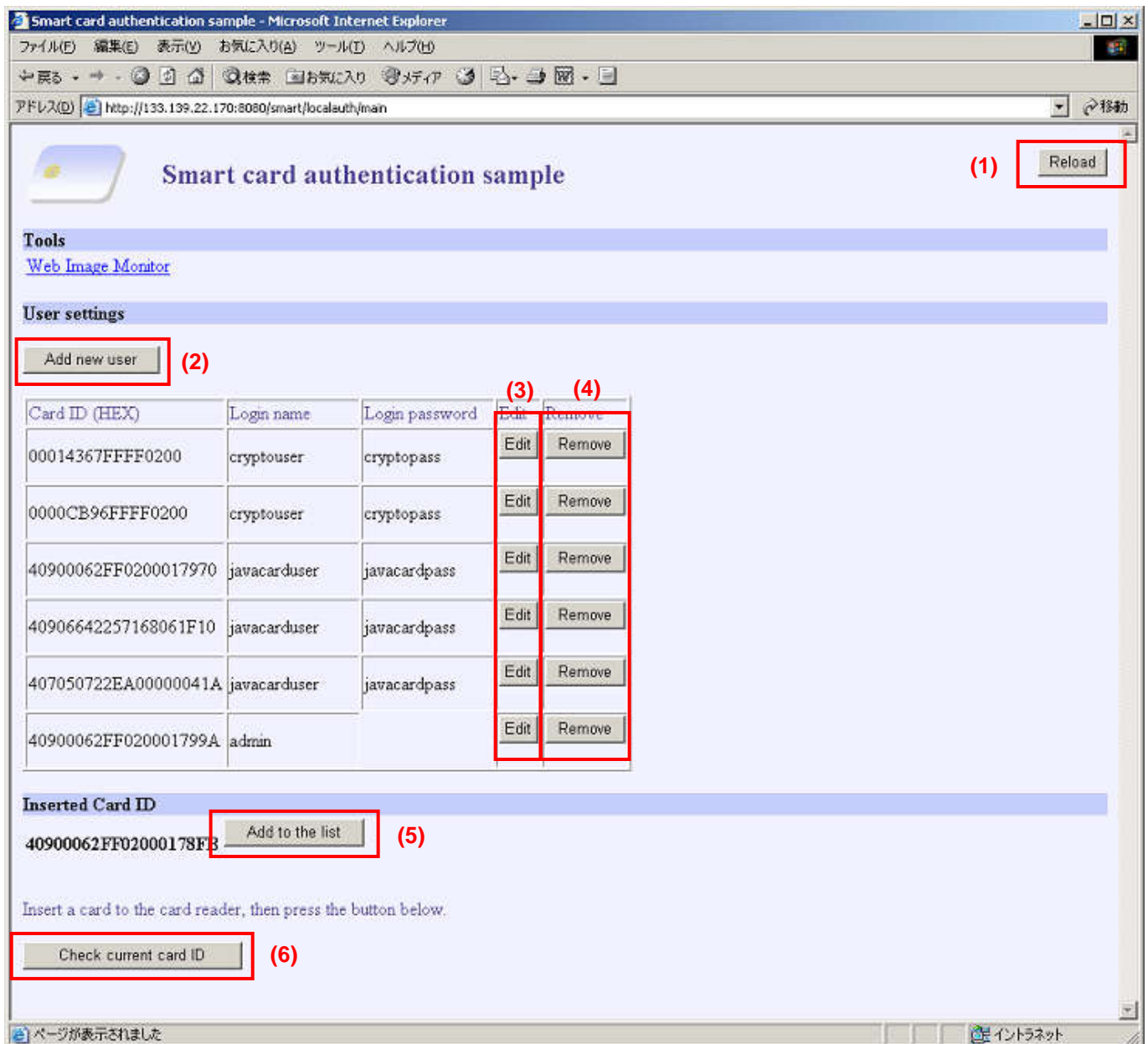
- Cryptoflex

4k, 8k, 8k v2, 16k, 32k, 32k v1, e-gate, e-gate 32k

10.3. Screen Images

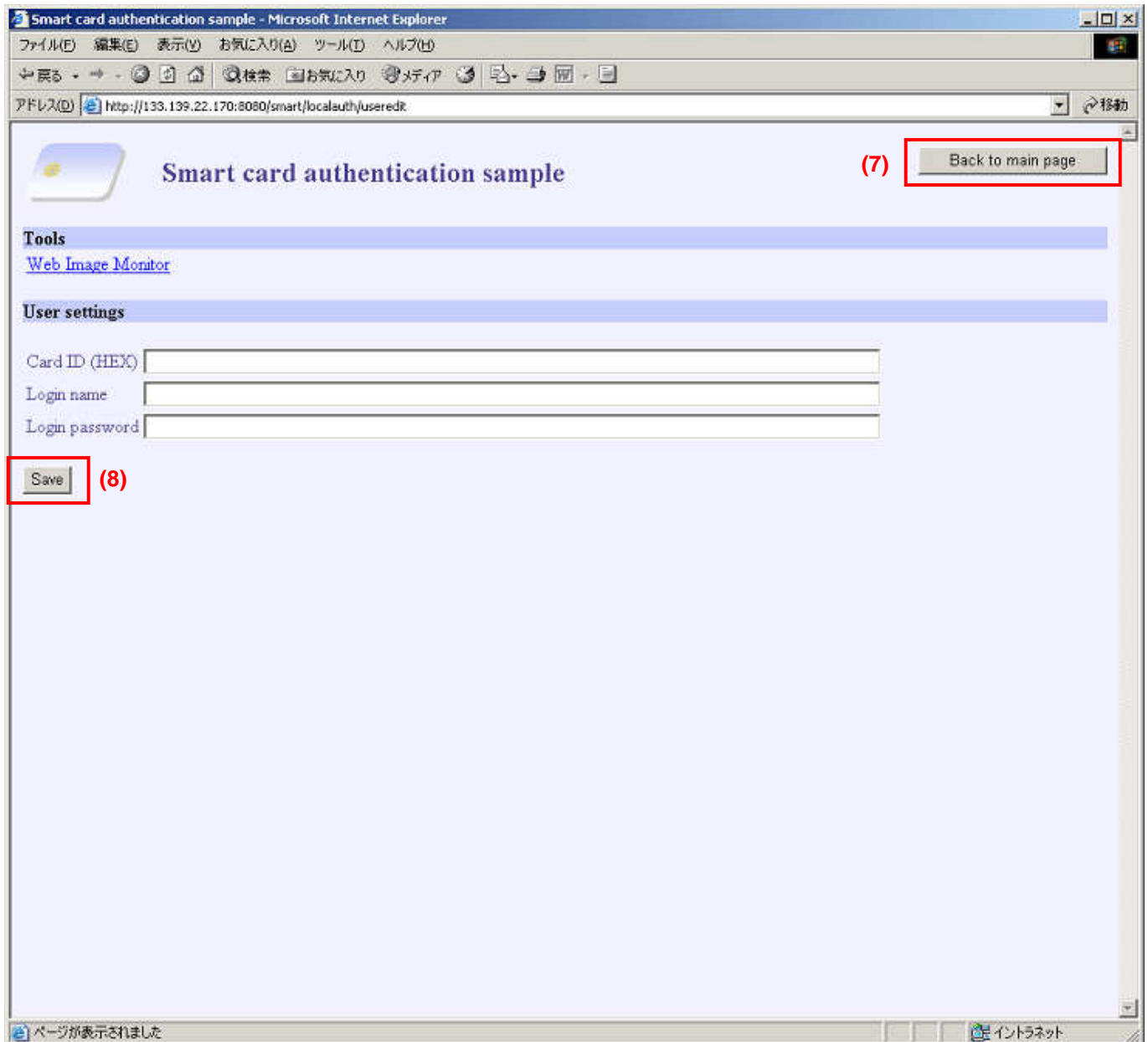
<<Web screen>>

- Main screen



No	Button	Action
(1)	Reload	Reloads the page.
(2)	Add new user	Adds a new ID to the ID list.
(3)	Edit	Edits the selected ID / login user name / login password.
(4)	Remove	Removes the selected ID.
(5)	Add to the list	Adds the displayed ID to the ID list.
(6)	Check current card ID	Obtains the currently inserted card ID and displays it on the screen.

- ID edit screen



No	Button	Action
(7)	Back to main page	Quits to edit the ID information and goes back to the main page.
(8)	Save	Updates the ID list and goes back to the main page.

<<Panel image>>

When the authentication management is set appropriately, the panel image will be as follows. (For 4 line LCD models, the panel image will be different one)

! Set an authentication card or enter login user name and login password, then press [Login].

(1) Login User Name
[Input Field]

Login Password
[Input Field]

(2) Enter
Enter

(3) Cancel

(4) Login

(1) Login User Name Display Area and Login Password Display Area

These areas display the login username and login password entered from the soft keyboard.

(2) Enter Buttons (Soft keyboard display buttons)

These buttons display a soft keyboard when they are clicked.

(3) Cancel Button

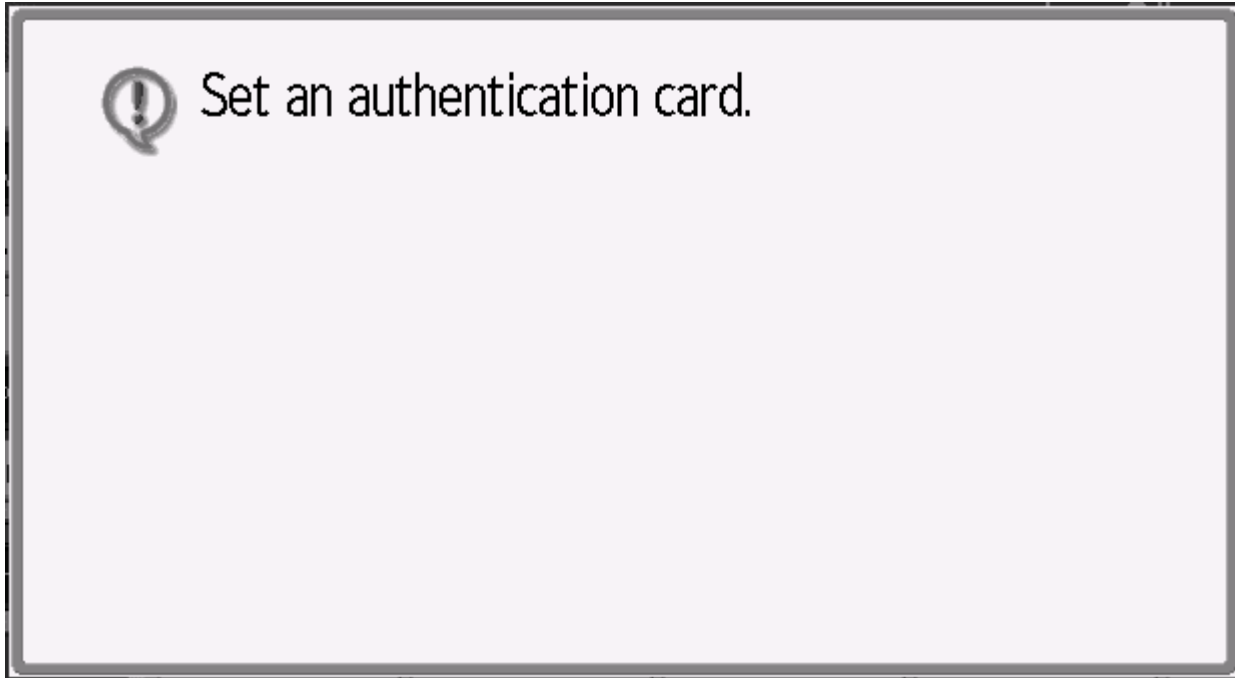
When this button is clicked, the entered user name and/or password are canceled.

(4) Login Button

When this button is clicked, a user authentication is performed with the given login user name and login password.

Note: This UI is for manual input authentication.

When only the card authentication is allowed in enhanced authentication management settings, the screen will be as follows. (For 4 line LCD models, the panel image will be different one)



10.4. Usage

Login

Set a card to the card reader device.

(For manual login, click the Enter button and enter a login user name and login password by using the soft keyboard.)

Logout

Press the [Login/Logout] hard-key when you are in login.

Edit the ID list

The ID list can be modified from web screen or by editing [INSTALLPATH]/home/localauth.dat file directly.

How to modify the ID list from web screen:

- Add a new ID

Press “Add new user” button or “Add to the list” button. Then, input each information and press “Save” button.

- Edit an ID and the corresponding login user name and login password

Press “Edit” button of the ID which you want to edit. Then, edit each information and press “Save” button.

- Remove an ID

Press “Remove” button of the ID which you want to remove.

11. Proximity card authentication sample

Location : sample/server/dist/285409978
URL : http://[ipaddress]:8080/proximity/localauth/main

11.1. Operation Environment

- PC/SC daemon is disabled. *
- Panel Service API is enabled. *
- No other SDK/J application using the Panel Service is currently active on the same device.
- The address book of the target MFP/LP has users whose login user name and the login password are set appropriately. (For details on the address book configuration, see the document supplied with the target device.)
- The basic user authentication management and the enhanced authentication management are enabled on the target device. (For details on the authentication function configuration, see the Panel Service User's Guide.)
- The card reader is configured properly.

* Please see "SDK/J Authentication Package Settings Guide" for details.

11.2. Features Overview

This sample application is a server type application using Panel Service. The work flow of this sample is as follows:

1. Detects a card.
2. Obtains theID of the detected card.
3. Converts the ID to the corresponding login user name and login password using the local ID list.
4. Performs the user authentication using the login user name and login password.

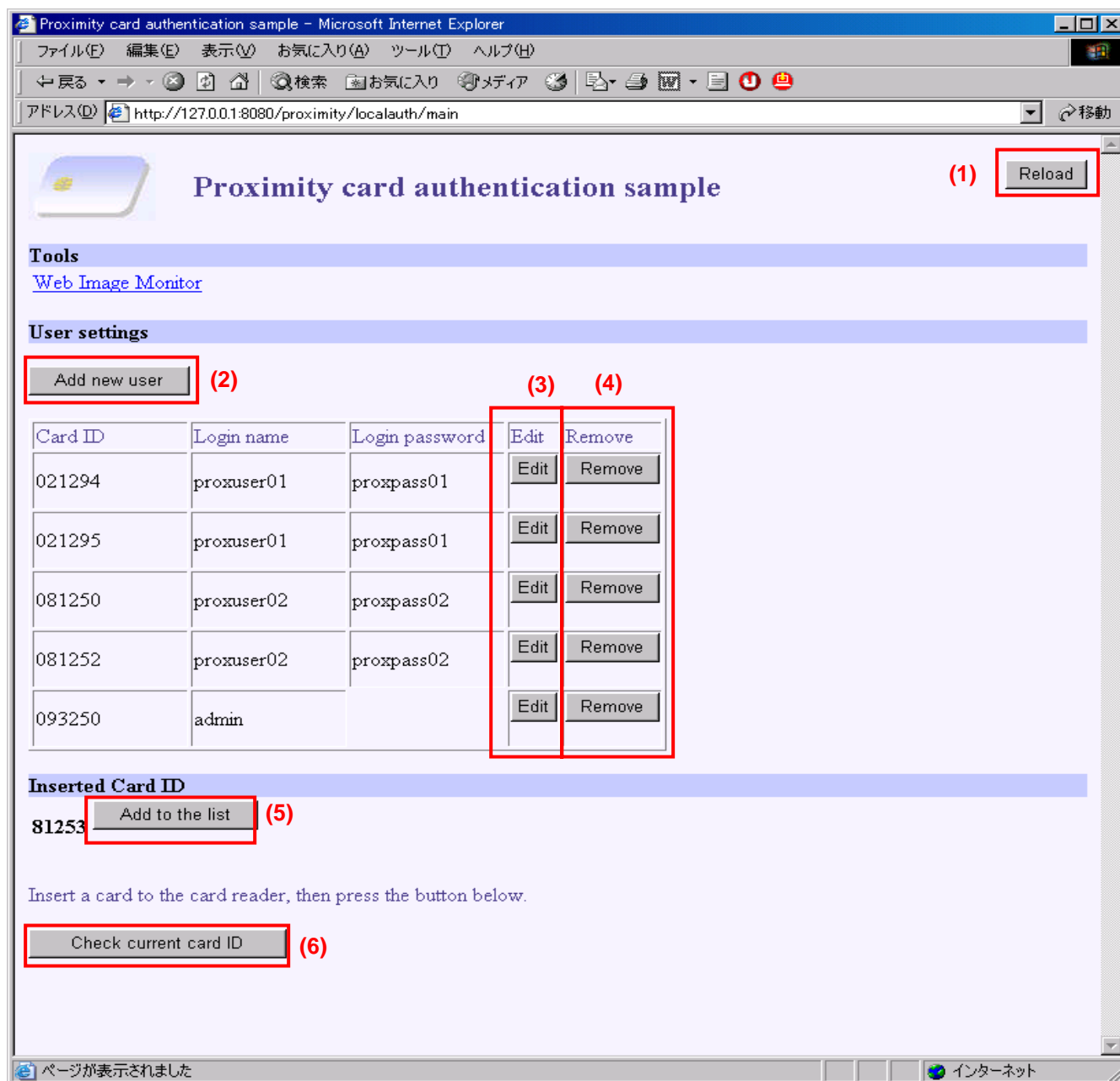
If the login user name and login password correspond which correspond to the detected card ID can not be found in the ID list, the user authentication will fail. The ID list can be modified from web screen or by editing [INSTALLPATH]/home/localauth.dat file directly.

This sample application uses javax.usb to communicate with cards. Please see "11.5. Card Reader Configuration" for the cards/card readers available in this sample application.

11.3. Screen Images

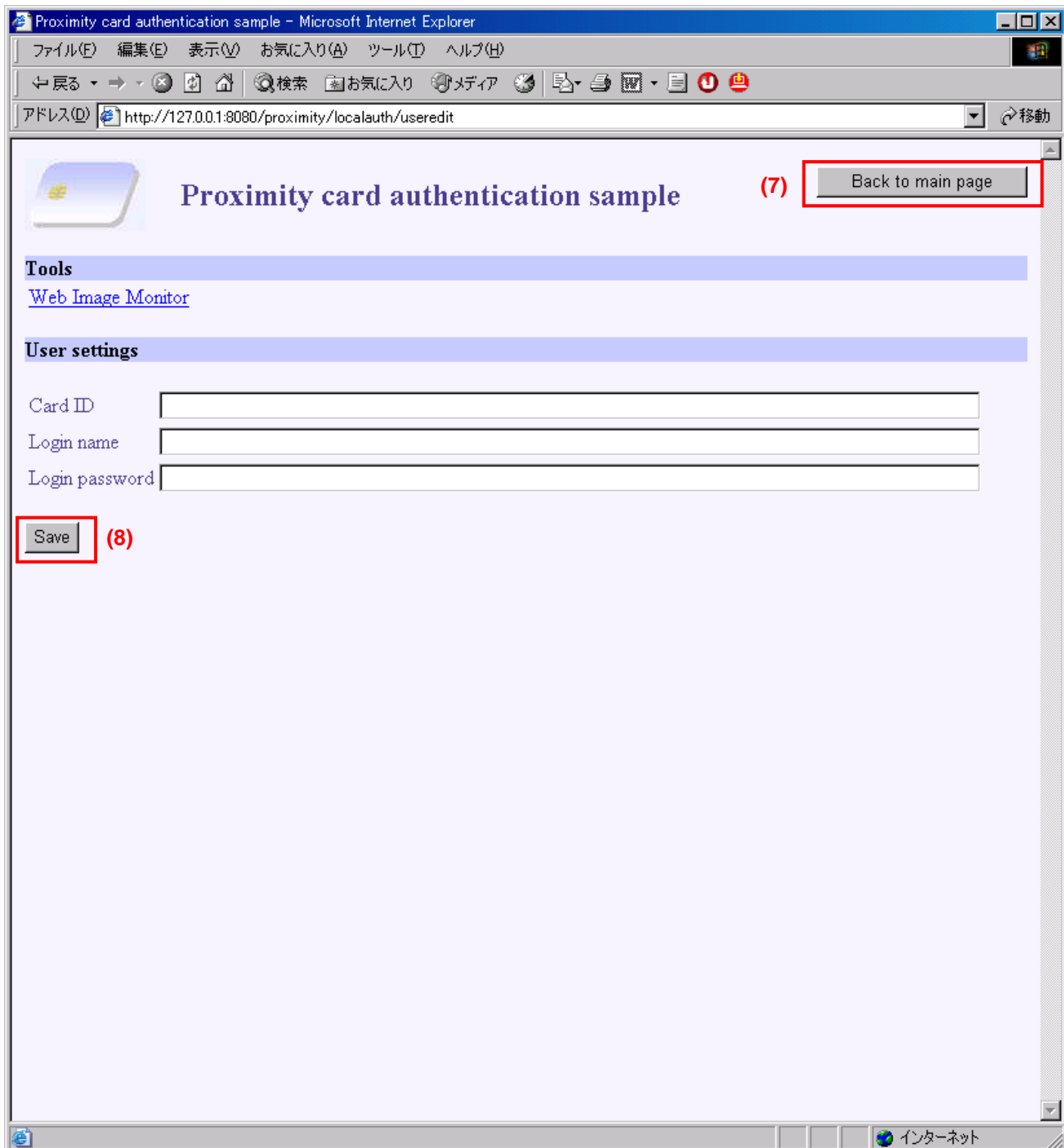
<<Web screen>>

- Main screen



No	Button	Action
(1)	Reload	Reloads the page.
(2)	Add new user	Adds a new ID to the ID list.
(3)	Edit	Edits the selected ID / login user name / login password.
(4)	Remove	Removes the selected ID.
(5)	Add to the list	Adds the displayed ID to the ID list.
(6)	Check current card ID	Obtains the previously detected card ID and displays it on the screen.

- ID edit screen



No	Button	Action
(7)	Back to main page	Quits to edit the ID information and goes back to the main page.
(8)	Save	Updates the ID list and goes back to the main page.

<<Panel image>>

When the authentication management is set appropriately, the panel image will be as follows. (For 4 line LCD models, the panel image will be different one)

(1) Login User Name Display Area and Login Password Display Area

These areas display the login username and login password entered from the soft keyboard.

(2) Enter Buttons (Soft keyboard display buttons)

These buttons display a soft keyboard when they are clicked.

(3) Cancel Button

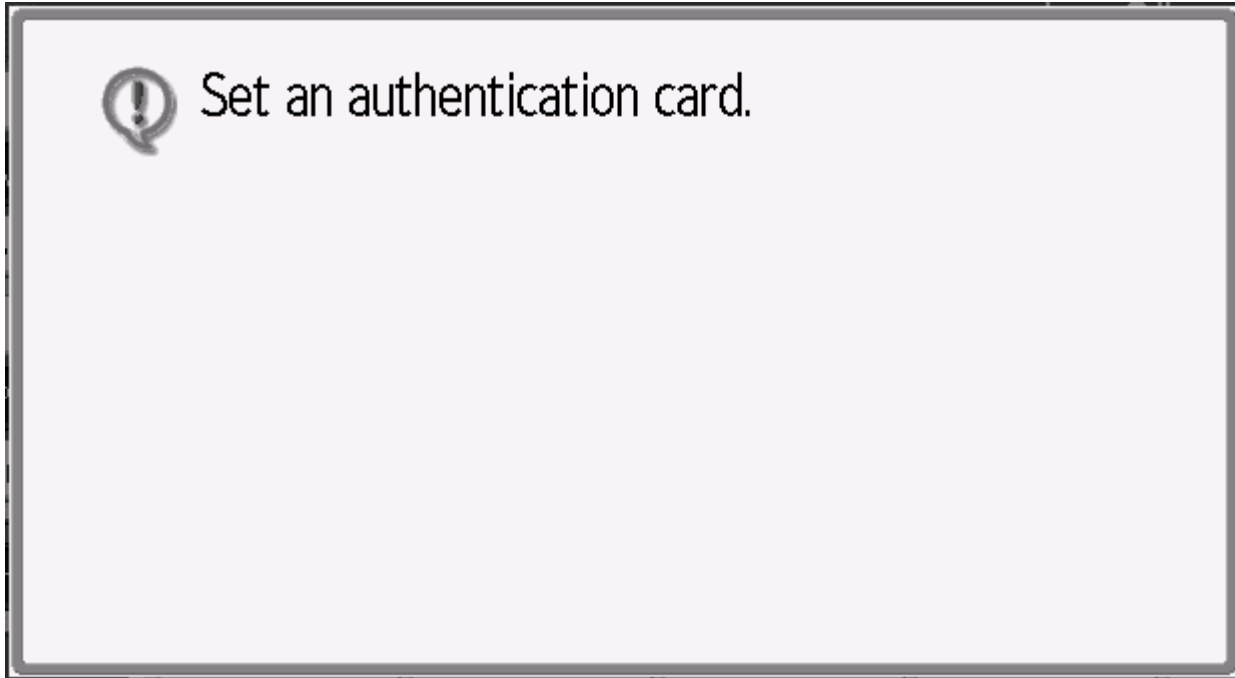
When this button is clicked, the entered user name and/or password are canceled.

(4) Login Button

When this button is clicked, a user authentication is performed with the given login user name and login password.

Note: This UI is for manual input authentication.

When only the card authentication is allowed in enhanced authentication management settings, the screen will be as follows. (For 4 line LCD models, the panel image will be different one)



11.4. Usage

Login

Set a card to the card reader device.

(For manual login, click the Enter button and enter a login user name and login password by using the soft keyboard.)

Logout

Press the [Login/Logout] hard-key when you are in login.

Edit the ID list

The ID list can be modified from web screen or by editing [INSTALLPATH]/home/localauth.dat file directly.

How to modify the ID list from web screen:

- Add a new ID

Press “Add new user” button or “Add to the list” button. Then, input each information and press “Save” button.

- Edit an ID and the corresponding login user name and login password

Press “Edit” button of the ID which you want to edit. Then, edit each information and press “Save” button.

- Remove an ID

Press “Remove” button of the ID which you want to remove.

11.5. Card Reader Configuration

The following table shows a list of the card readers supported by the sample applications. Before you use your card reader with this sample application, please configure the reader by following the instructions described in this section.

Card type	Manufacturer	Model	Description
HID	RFIDeas, Inc.	BSE-PCPRXH-U	Vendor ID : 0c27 Product ID : 3bfa Version : 0330
	RFIDeas, Inc.	RDR-6081AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0480
Indala/ Motorola	RFIDeas, Inc.	BSE-PCPRXM-U	Vendor ID : 0c27 Product ID : 3bfa Version : 0380
	RFIDeas, Inc.	RDR-6381AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0580
Casi-Rusco	RFIDeas, Inc.	RDR-6281AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0580
NexWatch	RFIDeas, Inc.	RDR-6N81AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0580
iClass Mifare(*1)	RFIDeas, Inc.	RDR-7081AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0582
Mifare(*1)	RFIDeas, Inc.	RDR-7581AKU	Vendor ID : 0c27 Product ID : 3bfa Version : 0560
Legic	interflex	IF 72 USB/RS232	Vendor ID : 0ce8 Product ID : 003b Version : 3469
Swipe	Tysso	TMSR-33-U-SB	Vendor ID : 1130 Product ID : 0001 Version : 0100

(*1)

ISO 14443, Type A - read only; MIFARE® Standard (serial number)

11.5.1. Configuring the working setting for the card reader

11.5.1.1. RFIDeas

1. Install the enroll tool

In order to configure the working setting for the card reader, first install the enroll tool on your PC.

The enroll tool can be obtained from: <http://www.rfideas.com/>.

This enroll tool is provided by RFIDeas, Inc. and can be used to configure the working setting for the card readers of that company.

2. Set the working setting for the card reader

Set the working setting for the card reader by using the enroll tool.

As the necessary working setting varies between models, be careful when configuring it.

[RDR-6081AKU, RDR-6381AKU]

Keystroke Data Tab

Advanced Tab

[RDR-6281AKU]

Keystroke Data Tab

pcProx and AIR ID Enroll Configuration Utility for USB and RS-232 Readers

Configuration Utility for pcProx® and AIR ID® Enroll

Timing Connect Card Formats **Set Keystroke Data** About Advanced

Facility (FAC) & ID Codes

PARITY BIT Strip parity bit count: Leading Parity 1

FACILITY CODE (FAC) ☒ Send FAC code ☐ Fac Hex

ID CODE ☒ Send ID Code Bit count of ID portion only 19

Force data to length ☒ FAC fixed to this length 6 ID fixed to this length 6

Extra keystroke/Character Sends

These pre-characters are sent ahead of card data: NOTE: Max of 3 total for pre and post keys. Pre-characters have priority.

☐ Enable FAC/ID character This char sent between FAC & ID COLON

These post-characters are sent after the card data: NONE NONE NONE

☐ Disable appending keystroke This keystroke appended to data ENTER

☐ Config changed ☐ Test RS-232 model

Use this field to view card data

Read pcProx or AIR ID OK

Write to pcProx or AIR ID Cancel

Advanced Tab

pcProx and AIR ID Enroll Configuration Utility for USB and RS-232 Readers

Configuration Utility for pcProx® and AIR ID® Enroll

Timing Connect Card Formats Set Keystroke Data **Advanced** About

LED - Beep Control

☐ SDK Controls LED ☒ Beep

☐ Red ☐ Green

1 USB Reader(s) on this Machine

Enter address 0-127

List of unique USB reader addresses 0 Change Reader

Software Developer Kit Mode

☐ Enables quiet mode for usage with the Software developer's Kit.

Raw Data

GET ID

Characters Sent When Card is Removed

First character

Second character

☐ Reverse Wiegand Bits ☐ Read only cards with this bit count 26

☒ Enable output as Hexadecimal ☐ Reverse Bytes

☒ Invert Wiegand Data (pcProxH only) ☒ Ignore HAW data inversion override

☐ Euro KeyPad ☐ Emulate ProxPro

☐ Enable 64 bit math

☐ Config changed ☐ Test RS-232 model

Use this field to view card data

Read pcProx or AIR ID OK

Write to pcProx or AIR ID Cancel

[RDR-6N81AKU]

Keystroke Data Tab

pcProx and AIR ID Enroll Configuration Utility for USB and RS-232 Readers

Configuration Utility for pcProx® and AIR ID® Enroll

Timing Connect Card Formats **Set Keystroke Data** About Advanced

Facility (FAC) & ID Codes

PARITY BIT Strip parity bit count: Leading Parity 0

FACILITY CODE (FAC) ☐ Send FAC code ☐ Fac Hex

ID CODE ☒ Send ID Code Bit count of ID portion only 26

Force data to length ☒ FAC fixed to this length 3 ID fixed to this length 3

Extra keystroke/Character Sends

These pre-characters are sent ahead of card data: NOTE: Max of 3 total for pre and post keys. Pre-characters have priority.

☐ Enable FAC/ID character This char sent between FAC & ID COLON

These post-characters are sent after the card data: NONE NONE NONE

☐ Disable appending keystroke This keystroke appended to data ENTER

☐ Config changed ☐ Test RS-232 model

Use this field to view card data 83718863

Read pcProx or AIR ID OK

Write to pcProx or AIR ID Cancel

Advanced Tab

pcProx and AIR ID Enroll Configuration Utility for USB and RS-232 Readers

Configuration Utility for pcProx® and AIR ID® Enroll

Timing Connect Card Formats Set Keystroke Data **Advanced** About

LED - Beep Control

☐ SDK Controls LED ☒ Beep

☐ Red ☐ Green

1 USB Reader(s) on this Machine

Enter address 0-127

List of unique USB reader addresses 0 Change Reader

Software Developer Kit Mode

☐ Enables quiet mode for usage with the Software developer's Kit.

Raw Data

GET ID

Characters Sent When Card is Removed

First character

Second character

☐ Reverse Wiegand Bits ☐ Read only cards with this bit count 26

☒ Enable output as Hexadecimal ☐ Reverse Bytes

☒ Invert Wiegand Data (pcProxH only) ☒ Ignore HAW data inversion override

☐ Euro KeyPad ☐ Emulate ProxPro

☐ Enable 64 bit math

☐ Config changed ☐ Test RS-232 model

Use this field to view card data 83718863

Read pcProx or AIR ID OK

Write to pcProx or AIR ID Cancel

[RDR-7081AKU]

This card reader supports both iClass and Mifare; set the working setting properly for the card you use.

IClass

Keystroke Data Tab

Advanced Tab

Mifare

Keystroke Data Tab

Advanced Tab

[RDR-7581AKU]

Keystroke Data Tab

Advanced Tab

This reader returns the reversed bytes of the actual card ID.

For example, if the card ID is “9A5F5026”, the card reader obtains it as “26505F9A”.

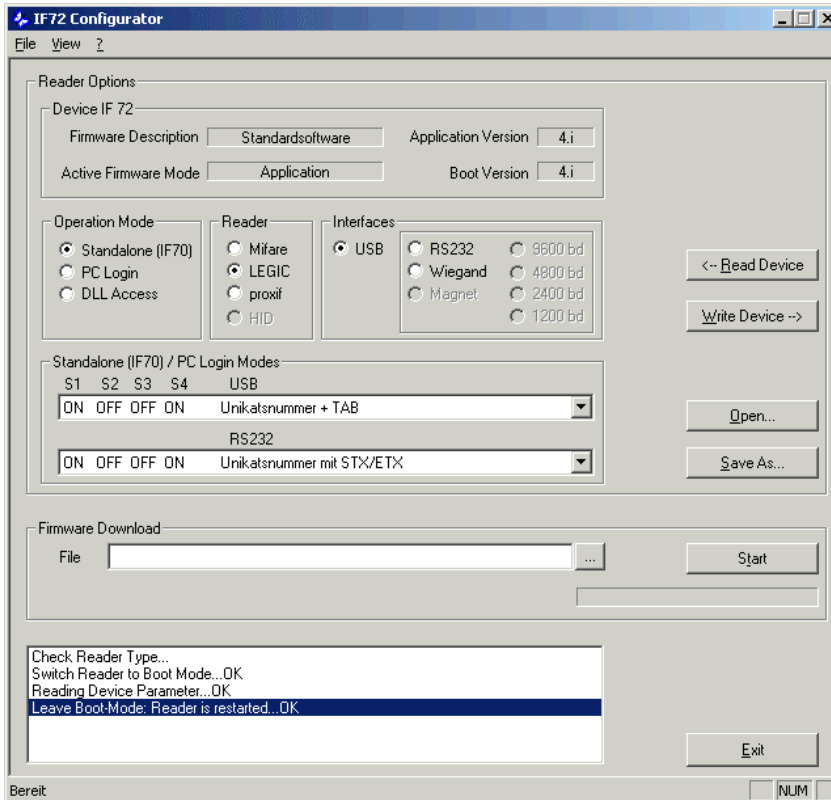
11.5.1.2. interflex

1. Install the enroll tool

In order to configure the working setting for the card reader, first install the enroll tool on your PC.
The enroll tool can be obtained from the CD-ROM. For details, see the manual of the card reader.

2. Set the working setting for the card reader

By using the enroll tool, set the working setting for the card reader as follows:



11.5.1.3. Tysso

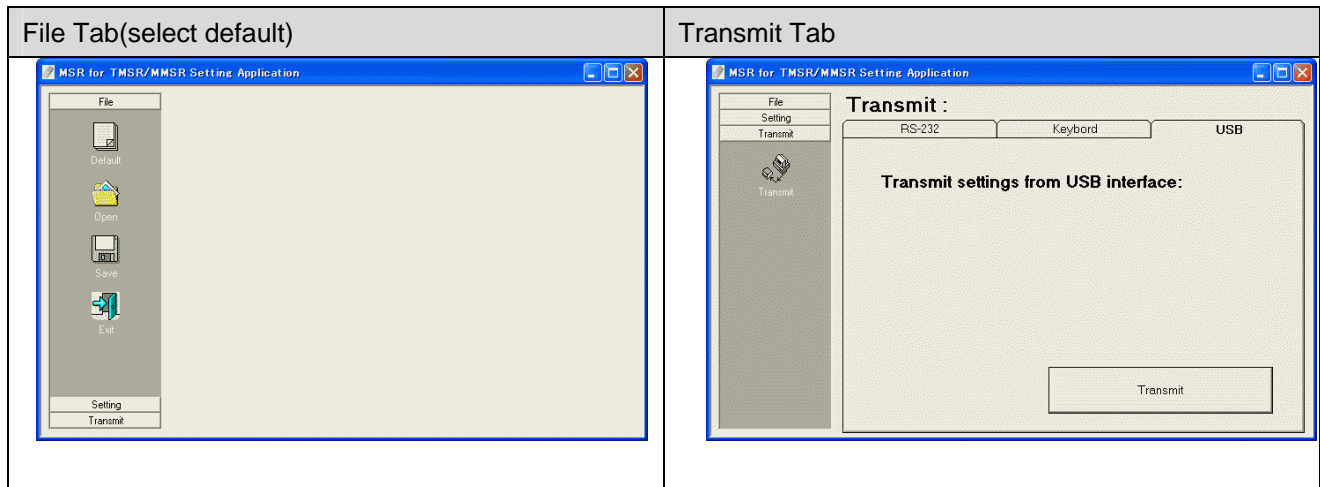
1. Install the enroll tool

In order to configure the working setting for the card reader, first install the enroll tool on your PC.

The enroll tool can be obtained from the CD-ROM. For details, see the manual of the card reader.

2. Set the working setting for the card reader

By using the enroll tool, set the working setting for the card reader. In this case, be sure to use the default setting.



11.5.2. Checking the operation of the card reader

Open the Notepad and check if the card ID is properly read.

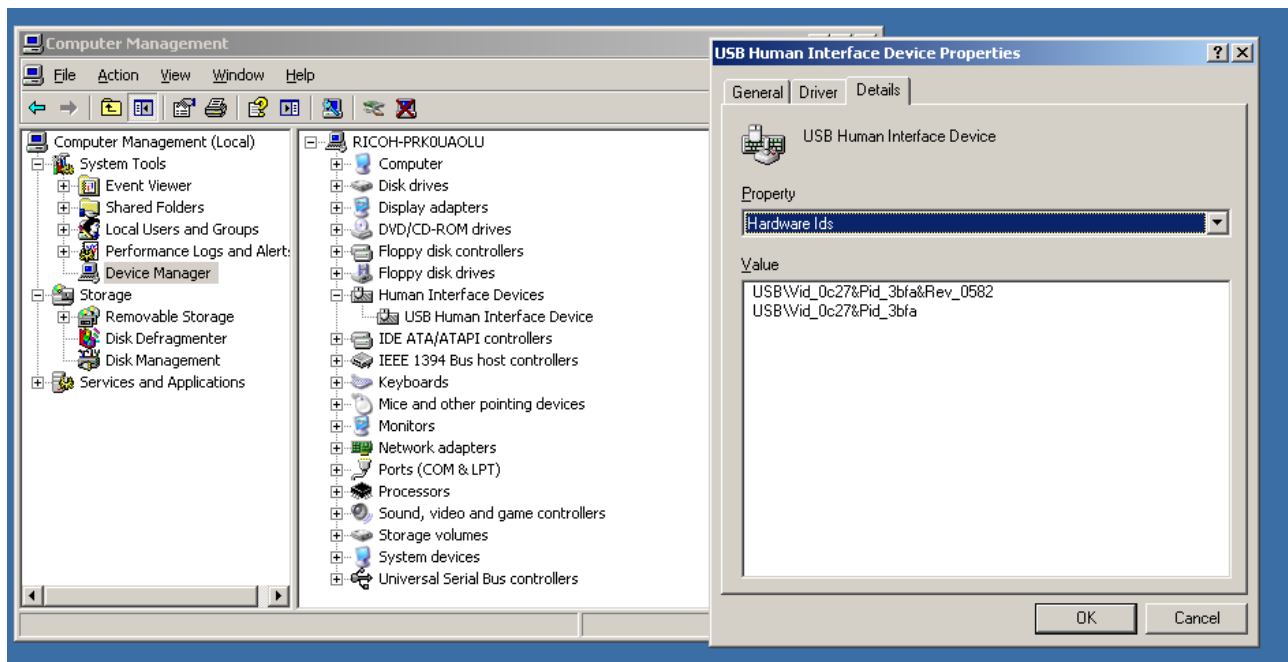
If a wrong value is obtained, check the working setting for the card reader.

11.5.3. Using another card reader than the supported card readers

If you wish to use a card reader that is not mentioned in “11.5 Card Reader Configuration”, you might first want to check the version of the card reader. You may be able to use the card reader for the sample applications if it can obtain the card ID in the following format: <Card ID><CR><LF>

1. Check the version of the card reader

1. Open **Device Manager**.
2. Select **Properties** in **Human Interface Devices**.
3. Select **Details** tab and select **Hardware Ids**.



When the value is “Vid_0c27&Pid_3bfa&Rev_0582”, it means:

Vendor ID=0c27

Product ID=3bfa

Release Version Number=0582

2. Register the information with the property file

Register the Vendor ID, Product ID, and Release Version Number with the property file.

The property file is available at: sdk/[dsdk or server]/dist/[product id]/card.reader.properties

Change History

Ver. 1.05	11.5.1.1. RFIDeas 1. Download Enroll Tool Modified the URL of the link.
Ver. 1.04	Update "Terms of Use and Trademarks"
Ver. 1.03	SDK/J v4 or later version based on SDK/J AP v1.0 option package for SDK/J v2. Update execution/operation environments for each application Integrate proximity sample and mifare sample Added some non pc/sc card readers support