



Value through Innovation

80088505-001-A

SmartPIN B100

JPOS User Manual

Revision History

Revision	Description	Date	By
A	Initial Release	1/17/2014	CH

Table of Contents

1.	Introduction.....	3
2.	System Requirements	3
3.	Reference Document	3
4.	Environment Build	3
5.	Properties, Methods, Events	5
6.	Run POSTest.....	8
7.	Sample Code.....	12

1. Introduction

This document provides information for using JPOS Service Object (SO) for SmartPIN B100 Device.

2. System Requirements

This JPOS Service Object is for Windows XP, Windows Vista and Windows 7.

3. Reference Document

1. Unified POS Retail Peripheral Architecture Version 1.13
2. International Standard for Implementation of Point Of Service Peripherals

4. Environment Build

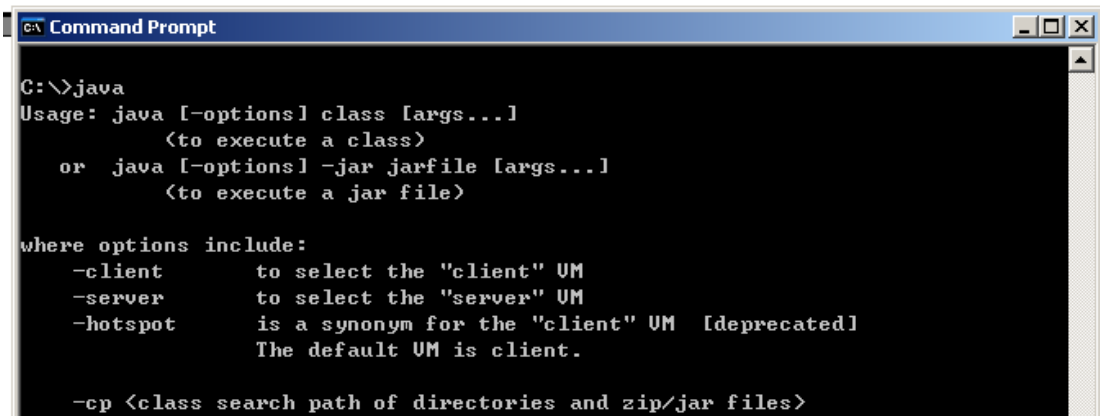
Install Sun JRE 1.5 or JDK 5.0, JRE 6 or JDK 6, JRE 7 or JDK 7:

Download the newest JRE 1.5 installation package from: <http://java.sun.com/j2se/1.5.0/download.jsp>, then click: [Download JDK 5.0 Update](#) and select a proper platform JDK version to download. Install the package and make sure the "java" command functions properly. There may be a need to add the JRE bin directory into the system environments variables by following steps:

For Microsoft Windows XP

1. Right click "My Computer" in the desktop
2. Select the "properties" menu item
3. Select "Advanced" tab
4. Click button "Environment Variables"
5. Select "Path" and insert the JRE bin directory.

When Java is working, following can be observed:



```

c:\ Command Prompt
C:\>java
Usage: java [-options] class [args...]
        (to execute a class)
    or java [-options] -jar jarfile [args...]
        (to execute a jar file)

where options include:
    -client    to select the "client" UM
    -server    to select the "server" UM
    -hotspot   is a synonym for the "client" UM [deprecated]
               The default UM is client.

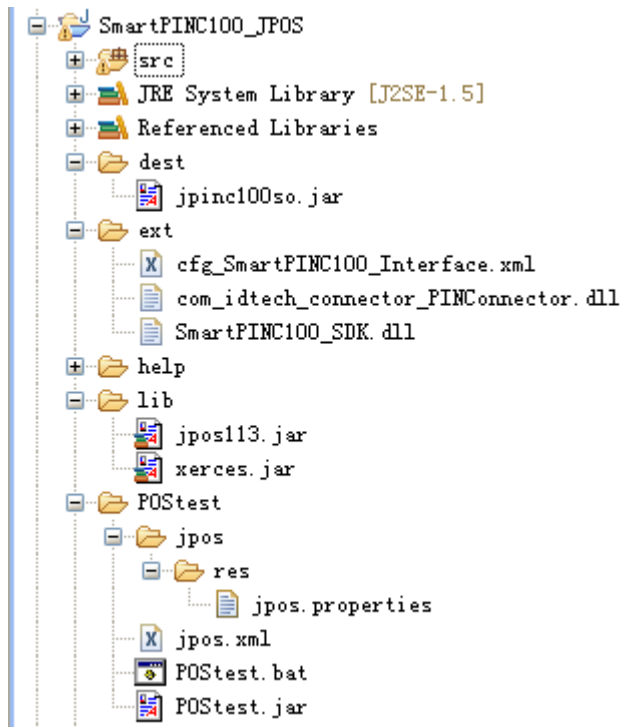
    -cp <class search path of directories and zip/jar files>

```

Enter DOS command shell, and enter java. It shows information about how to use.

FILELISTS:

Below frame structure comes from the IDE –eclipse:



jpinc100so.jar – JPOS device service control for SmartPIN B100 Device

com_idtech_connector_PINConnector.dll – PINPad JNI DLL

SmartPINB100_SDK.dll – SmartPIN B100 SDK, extension DLL to access USBHID or RS232

cfg_SmartPINB100_Interface.xml – Interface configure file.

The .DLL and .XML files should be located in the directory in the PATH environment specified. For example, if the current working directory is included in the PATH variable:

```
PATH=%PATH%;
```

So, the files can be moved to current directory.

jpos.xml – jpos configure file, jpos register file. There are several entries in this file, with entry name:

logicalName

“defaultPINPad”

“SmartPIN_B100_USBHID”

“SmartPIN_B100_RS232”

Specific properties in the jpos.xml:

“DeviceClass” – specify the class of current using device;

“DeviceName” – specify the interface of current using device;

“SDK” – specify the communication SDK of current using device.

“defaultPINPad”– this entry is a copy of one of other entry as the default name for POSTest program.

“SmartPIN_B100_USBHID” –

“DeviceClass”: “PINPad”

“DeviceName”: “USBHID”

“SDK”: “SmartPINB100_SDK.dll”

“SmartPIN_B100_RS232” –

“DeviceClass”: “PINPad”

“DeviceName”: “RS232”

“SDK”: “SmartPINB100_SDK.dll”

Jpos/res/jpos.properties – jpos configure file, this specifies jpos.xml

This file specify which configure file is used. By default, it specifies the “jpos.xml” as the one.

Add jsignpayso.jar to your CLASSPATH or specify it in the javac / java command.

set CLASSPATH=%CLASSPATH%;yourDir\jsignpayso.jar

For example: jsignpayso.jar is placed in c:\so,

it should set CLASSPATH=%CLASSPATH%;c:\so\jsignpayso.jar

5. Properties, Methods, Events

Properties:

Please see OPOS PINPad Spec if the detailed information is wanted.

Name	Type	Mutability	Supported?
AutoDisable	boolean	read-write	not support
CapCompareFirmwareVersion	boolean	read-only	not support
CapPowerReporting	int32	read-only	not support
CapStatisticsReporting	boolean	read-only	not support
CapUpdateFirmware	boolean	read-only	not support
CapUpdateStatistics	boolean	read-only	not support
CheckHealthText	string	read-only	support
Claimed	boolean	read-only	support
DataCount	int32	read-only	support
DataEventEnabled	boolean	read-write	support
DeviceEnabled	boolean	read-write	support
FreezeEvents	boolean	read-write	support
OutputID	int32	read-only	not support
PowerNotify	int32	read-write	not support
PowerState	int32	read-only	not support
State	int32	read-only	support
DeviceControlDescription	string	read-only	support
DeviceControlVersion	int32	read-only	support
DeviceServiceDescription	string	read-only	support
DeviceServiceVersion	int32	read-only	support
PhysicalDeviceDescription	string	read-only	support
PhysicalDeviceName	string	read-only	support
CapDisplay	int32	read-only	support
Capkeyboard	boolean	read-only	support
CapLanguage	int32	read-only	support
CapMACCalculation	boolean	read-only	not support

CapTone	boolean	read-only	support
AccountNumber	string	read-write	support
AdditionalSecurityinformation	string	read-only	support
Amount	currency	read-write	support
AvailableLanguagesList	string	read-only	Not support
AvailablePromptsList	string	read-only	not support
EncryptedPIN	string	read-only	support
MaximumPINLength	int32	read-write	support
MinimumPINLength	int32	read	support
MerchantID	string	read-only	support
PINEntryEnabled	boolean	read-only	support
Prompt	int32	read-only	Not support
PromptLanguage	nls	read-only	Not support
TerminalID	string	read- write	support
Track1Data	binary	read-write	not support
Track2Data	binary	read-write	not support
Track3Data	binary	read-write	not support
Track4Data	binary	read-write	not support
TransactionType	string	read-only	support

Methods:

Please see OPOS PINPad Spec if the detailed information is wanted.

Name	Supported?
Open	support
Close	support
Claim	support
Release	support
Checkhealth	support
clearInput	support
clearOutput	not support
Direction	support
compareFirmwareVersion	not support
resetStatistics	not support
retrivevStatistics	not support
updateFirmware	not support
updateStatistics	not support
beginEFTTransaction	support
CapMAC	not support
enablePINEntry	support
endEFTTransaction	support
updateKey	not support

verifyMAC	not support
-----------	-------------

Open (LPCTSTR DeviceName)

Parameter: DeviceName, the "SmartPIN_B100_RS232" for RS232 interface device, and the "SmartPIN_B100_USBHID" for USBHID interface device.

Return: OPOS_SUCCESS if every thing works, or OPOS error code (see UPOS Specification).

Events:

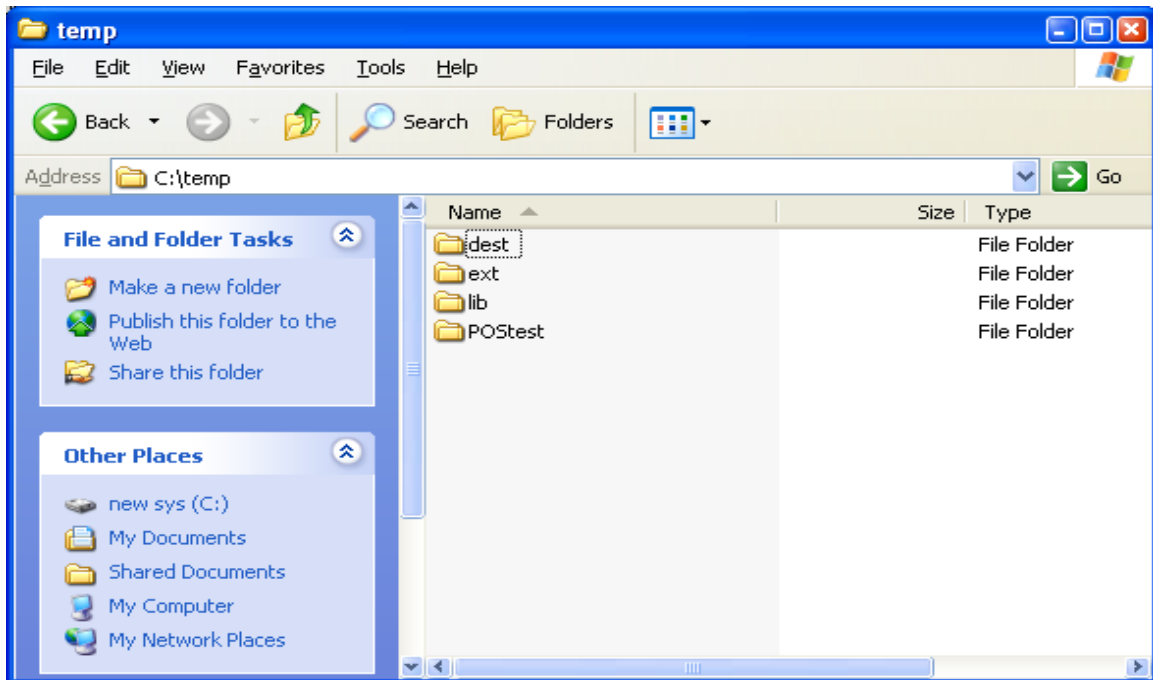
These events are fired by the Service Object when it is necessary. The following functions are, in fact, the event-handlers that can be added into the applications. Then the applications can receive these events and do some processing accordingly. Please see OPOS PINPad Spec if the detailed information is wanted.

Name	Usage
DataEvent	support
ErrorEvent	support
DirectIOEvent	support
OutputCompleteEvent	not support
StatusUpdateEvent	not support

6. Run POSTest

It's the official recommended JPOS compatible test program. The release version includes the program or it can be gotten from <http://sourceforge.net/projects/postest/>

1. Unzip file "SmartPINB100_JPOS.zip" to a directory. Such as c:\temp, The Directory is like the following:



Enter "c:\temp\Postest" directory under the DOS command shell

```
C:\WINDOWS\System32\cmd.exe

C:\temp>dir
Volume in drive C is NEW SYS
Volume Serial Number is AC64-BBF1

Directory of C:\temp

03/21/2006  11:06 AM    <DIR>          .
03/21/2006  11:06 AM    <DIR>          ..
05/16/2006  11:04 AM    <DIR>          dest
05/16/2006  11:04 AM    <DIR>          ext
05/16/2006  11:04 AM    <DIR>          lib
05/16/2006  11:04 AM    <DIR>          POSTest
               0 File(s)                0 bytes
               6 Dir(s)  5,738,045,440 bytes free

C:\temp>
```

Then enter subdirectory "POSTest"


```

C:\WINDOWS\System32\cmd.exe

C:\temp>dir
Volume in drive C is NEW SYS
Volume Serial Number is AC64-BBF1

Directory of C:\temp

03/21/2006  11:06 AM    <DIR>          .
03/21/2006  11:06 AM    <DIR>          ..
05/16/2006  11:04 AM    <DIR>          dest
05/16/2006  11:04 AM    <DIR>          ext
05/16/2006  11:04 AM    <DIR>          lib
05/16/2006  11:04 AM    <DIR>          POSTest
                0 File(s)            0 bytes
                6 Dir(s)   5,738,045,440 bytes free

C:\temp>cd postest

C:\temp\POSTest>

```

2. To run POSTest just like following:

```

C:\WINDOWS\System32\cmd.exe

C:\temp>cd postest

C:\temp\POSTest>dir
Volume in drive C is NEW SYS
Volume Serial Number is AC64-BBF1

Directory of C:\temp\POSTest

05/19/2006  08:53 AM    <DIR>          .
05/19/2006  08:53 AM    <DIR>          ..
05/16/2006  11:04 AM                761 jposps2.xml
05/16/2006  11:04 AM                799 jposserial.xml
05/16/2006  11:04 AM                774 jposusbhid.xml
05/16/2006  11:04 AM                773 jposusbkb.xml
05/16/2006  01:38 PM                186 POSTest.bat
05/16/2006  11:04 AM            322,874 POSTest.jar
05/16/2006  11:04 AM    <DIR>          jpos
05/16/2006  11:04 AM                128 vssver.scc
05/16/2006  11:04 AM                187 POSTest.bat.bak
                8 File(s)            326,482 bytes
                3 Dir(s)   5,738,045,440 bytes free

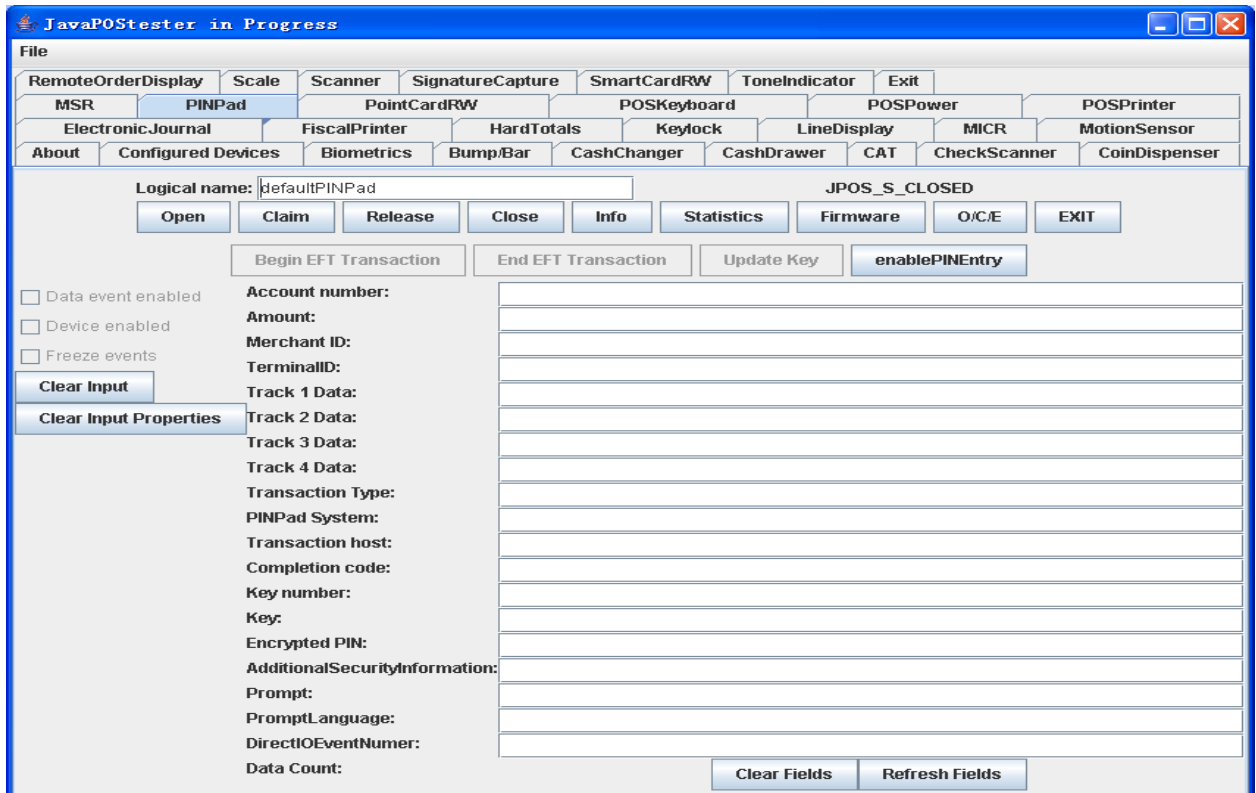
C:\temp\POSTest>postest

```

Then a POSTester program screen is shown.

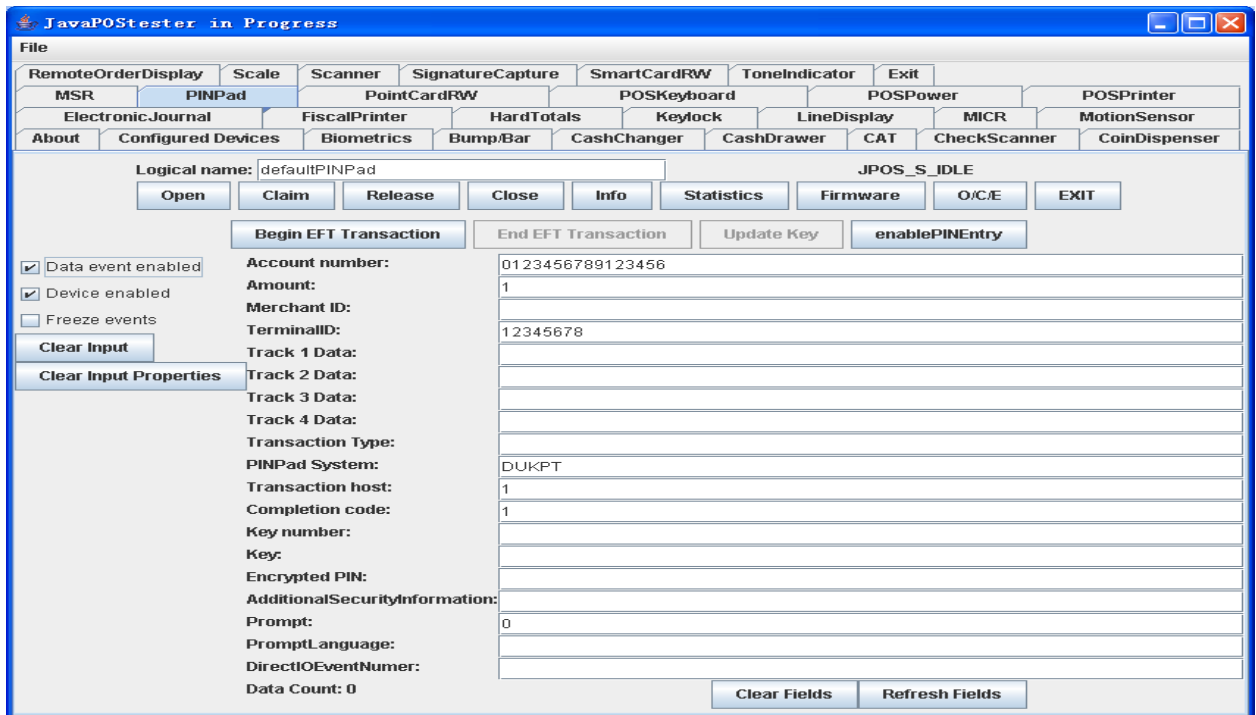


3. Click "PINPad" Tab

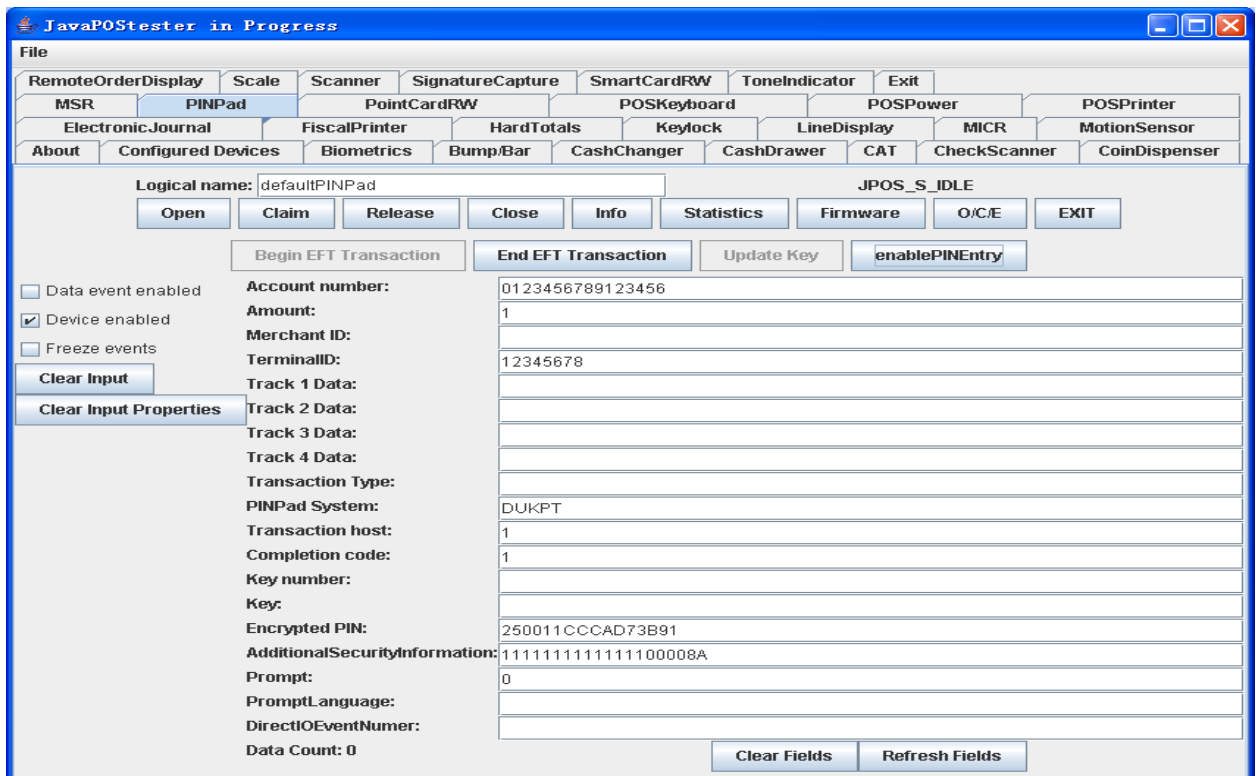


The logical name is the entry name in the jpos.xml file. Click "Open" button, and "Claim", and check choices "Data event enabled", "Device enabled":

JPOS Service Object Reference, SmartPIN B100



Click "Begin EFT Transaction" and "enablePINEntry", and then enter the PIN through the device:



Click "End EFT Transaction", and click "Release" or "Close" to release or close the JPOS,

NOTE: This should be done to make sure the device returns to normal state.

Click "EXIT" to quit the program.

For more detail please check the <http://www.javapos.com> and the release help files.

7. Sample Code

The following code is the sample of PINPad:

```
import jpos.*;
import jpos.config.*;
import jpos.config.simple.*;
import jpos.loader.*;
import jpos.loader.simple.*;
import jpos.profile.*;
import jpos.services.*;
import jpos.util.*;
import jpos.util.tracing.*;

public class YourClass{

    public static void main(String [] args){
        PINPad pin=new PINPad(); /* new an instance */

        try{
            pin.open(logicalName);
            pin.claim(100);
            pin.setDeviceEnabled (true);
        }catch(JposException e){
            System.err.print(e);
        }
        /**
         * Do something here.
         */
        try{
            pin.release();
            pin.close();
        }catch(JposException e){
            System.err.print(e);
        }
    }
}
```