IDTECH™

Value through Innovation

USER MANUAL

SecuRED

Encrypted Magstripe Reader

JPOS Reference Guide

Windows

80128504-001

Rev A. 9/9/13

SecuRED JPOS User Manual

Revision History

Revision	Date	Description	By
A	9/9/2013	Initial Release	CH

Table of Contents

1. Introduction

This document provides information for using JPOS Service Object (SO) for MSR readers.

2. System Requirements

This JPOS Service Object is for Windows XP, Windows Vista, Windows 7 and Windows 8.

3. Reference Document

1. Unified POS Retail Peripheral Architecture Version 1.13.1
2. International Standard for Implementation of Point Of Service Peripherals
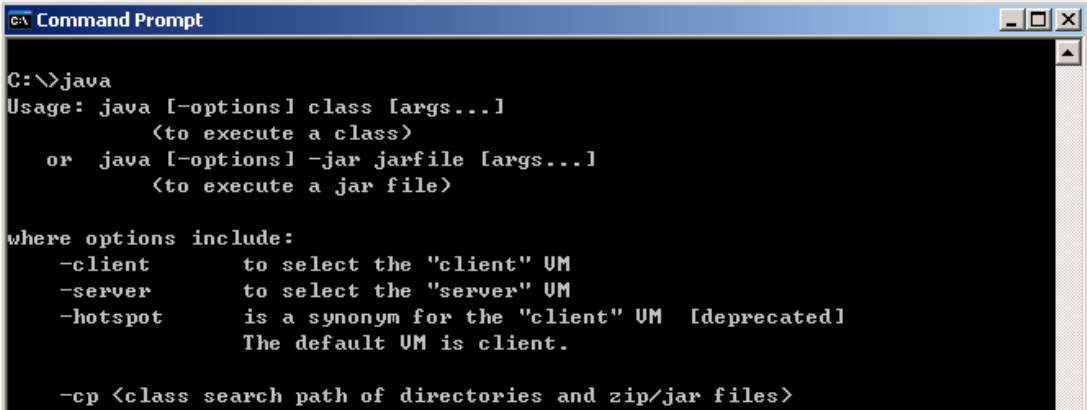
4. Environment Build

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.

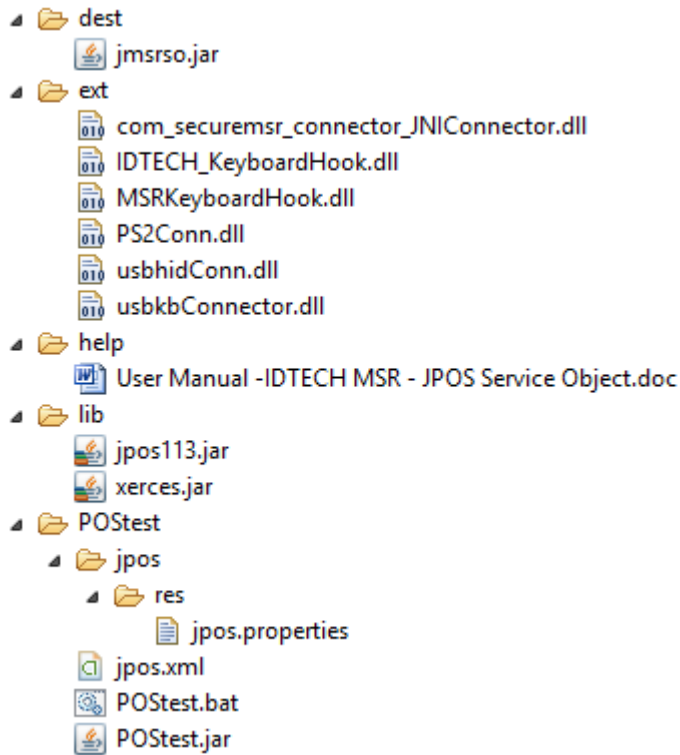When Java is working, following can be observed:

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

FILELISTS:

jmsrso.jar - JPOS device service control for MSR

jpos.xml - jpos configure file

jpos.properties - jpos configure file, this specifies jpos.xml

com_securemsr_connector_JNIConnector.dll - JNI DLL

usbkbConnector.dll - extension DLL

kbhidConn.dll - wrap DLL to usbhid and kbhid

usbhidConn.dll - extension DLL to access usbhid

MSRKeyboardHook.dll - extension DLL

The .DLL 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 DLLs can be moved to current directory.

Device Name: the jpos.xml the JPOS register file, which specify the device names.

There are several entries in this file, with entry name:

    "defaultMSR"

    "IDTECHSECURED_USBHID"

    "IDTECHSECURED_USBKB"

    "IDTECHSECUREMAG_USBHID"

    "IDTECHSECUREMAG_USBKB"

    "IDTECHMINIMAG_USBHID

    "IDTECHMINIMAG_USBKB"

"IDTECHVALUEMAG_USBHID"

"IDTECHVALUEMAG_USBKB"

"IDTECHMSR_PS2KB"

"IDTECHMSR_RS232"

"IDTECHMSR_USBCDC"

"defaultMSR"

The entry is a copy of one of other entry as the default name for POSTest program.

"IDTECHSECURED_USBHID",

The line: <prop name="connector" type="String" value=" usbhid/0acd/2810 "/> specifies the connector driver name, and the configuration value. 0acd is IDTECH Vendor ID; 2810 is the product IDs

jpos/res/jpos.properties

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

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

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

　　　For example: jmsrso.jar is placed in c:\so, it should set CLASSPATH=%CLASSPATH%;c:\so\jmsrso.jar
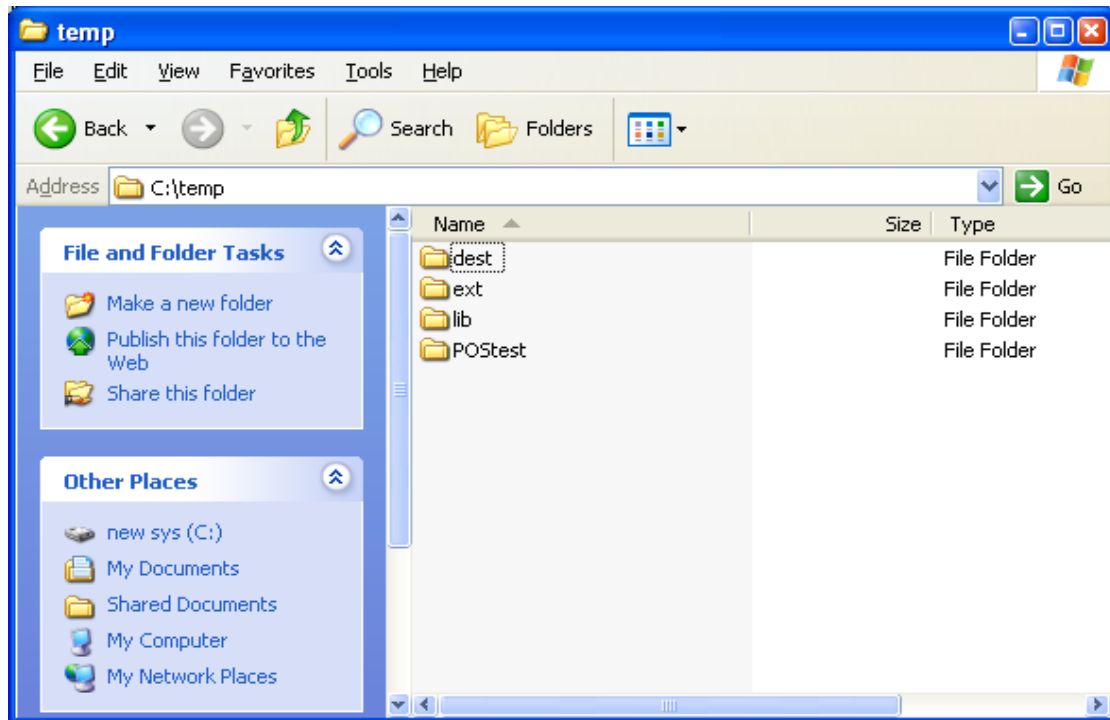
5. Run POSTest

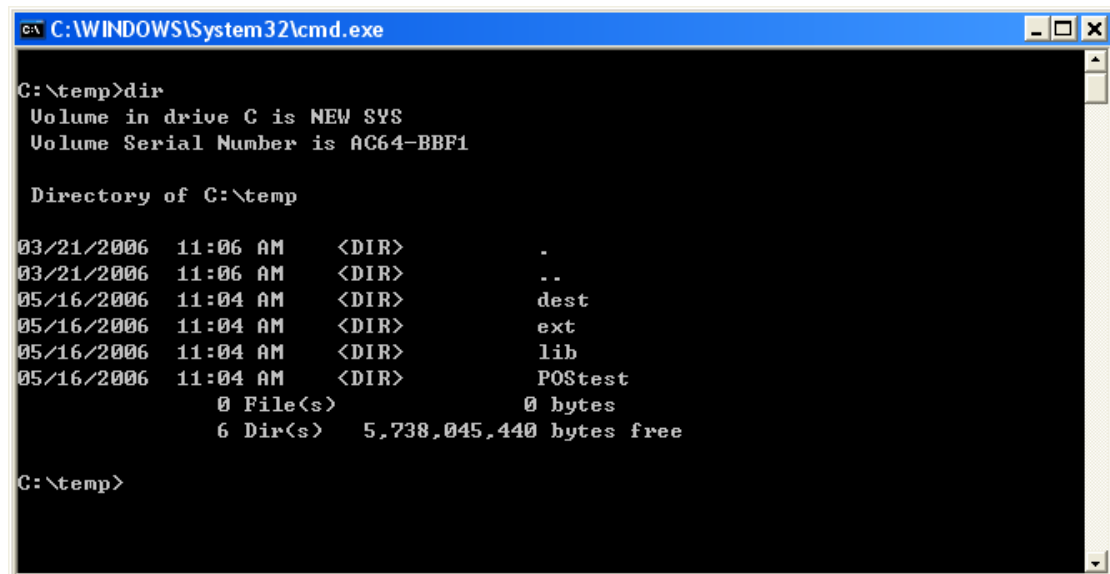It's the official recommended JPOS compatible test program. The release version includes the program or it can be gotten from http://sourceforge.net/projects/postest/

1. Unzip file to a directory. Such as

　　　c:\temp

The Directory is like the following:

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

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>
```
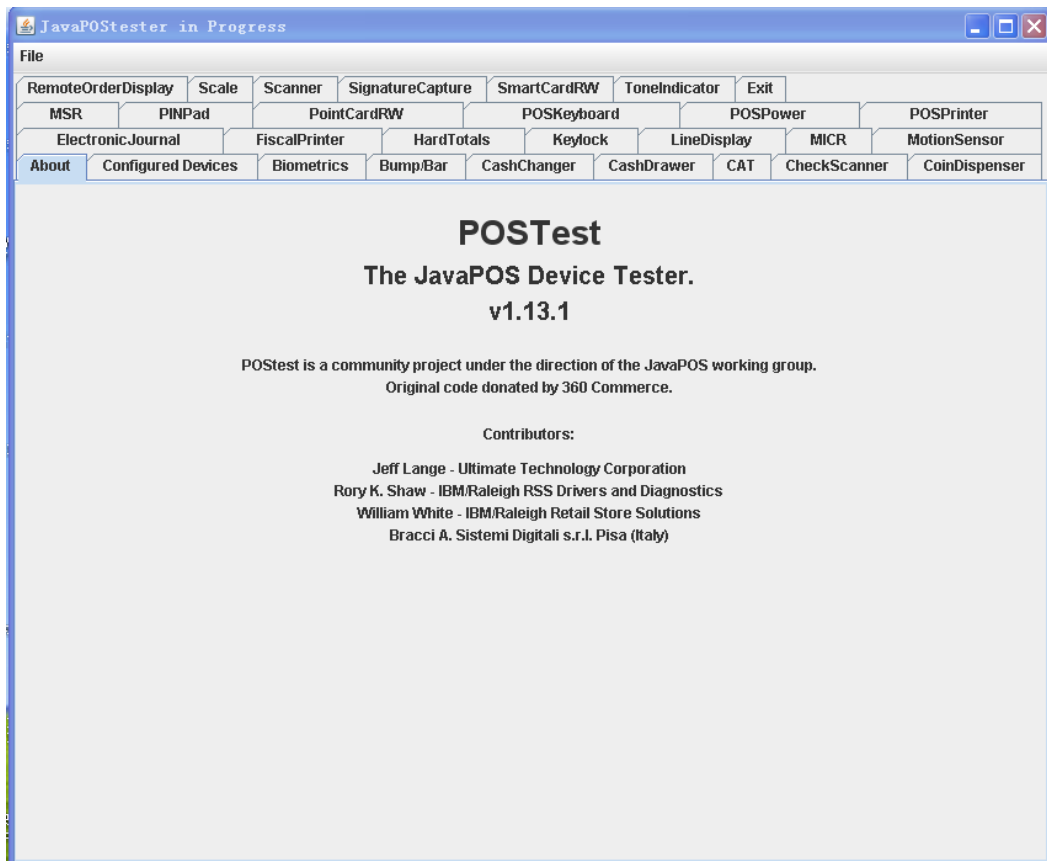
2. To run POSTest just like following:

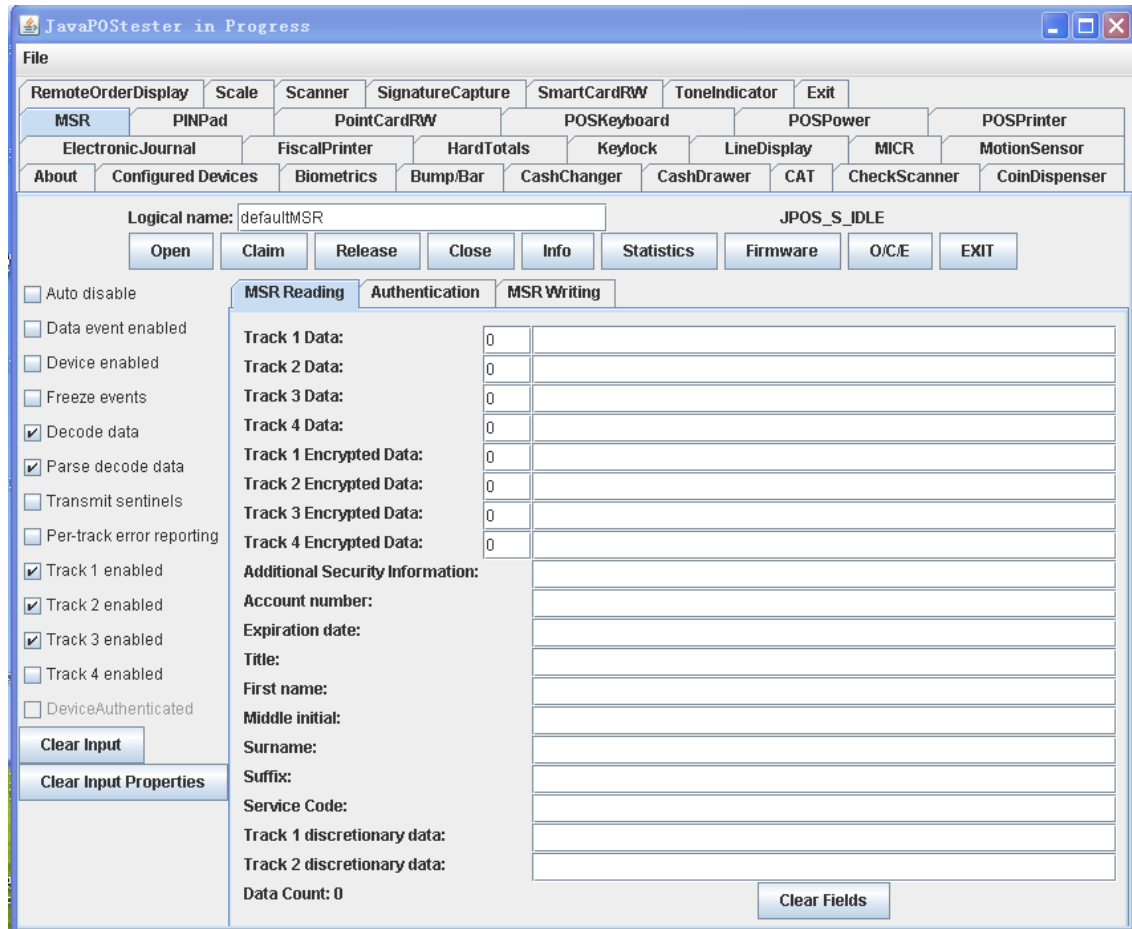Then a POSTest program screen is shown.

3. Click "MSR" Tab
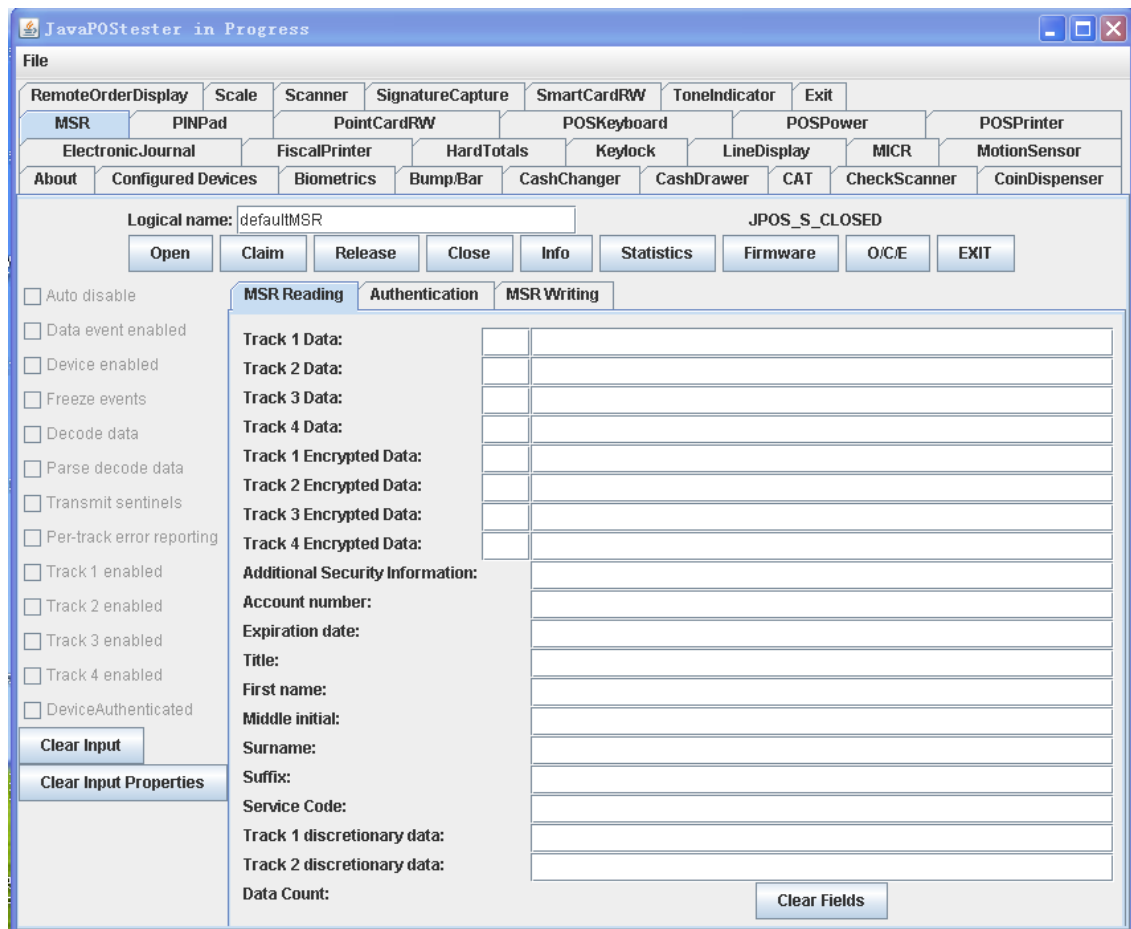
The logical name is the entry name in the jpos.xml file.
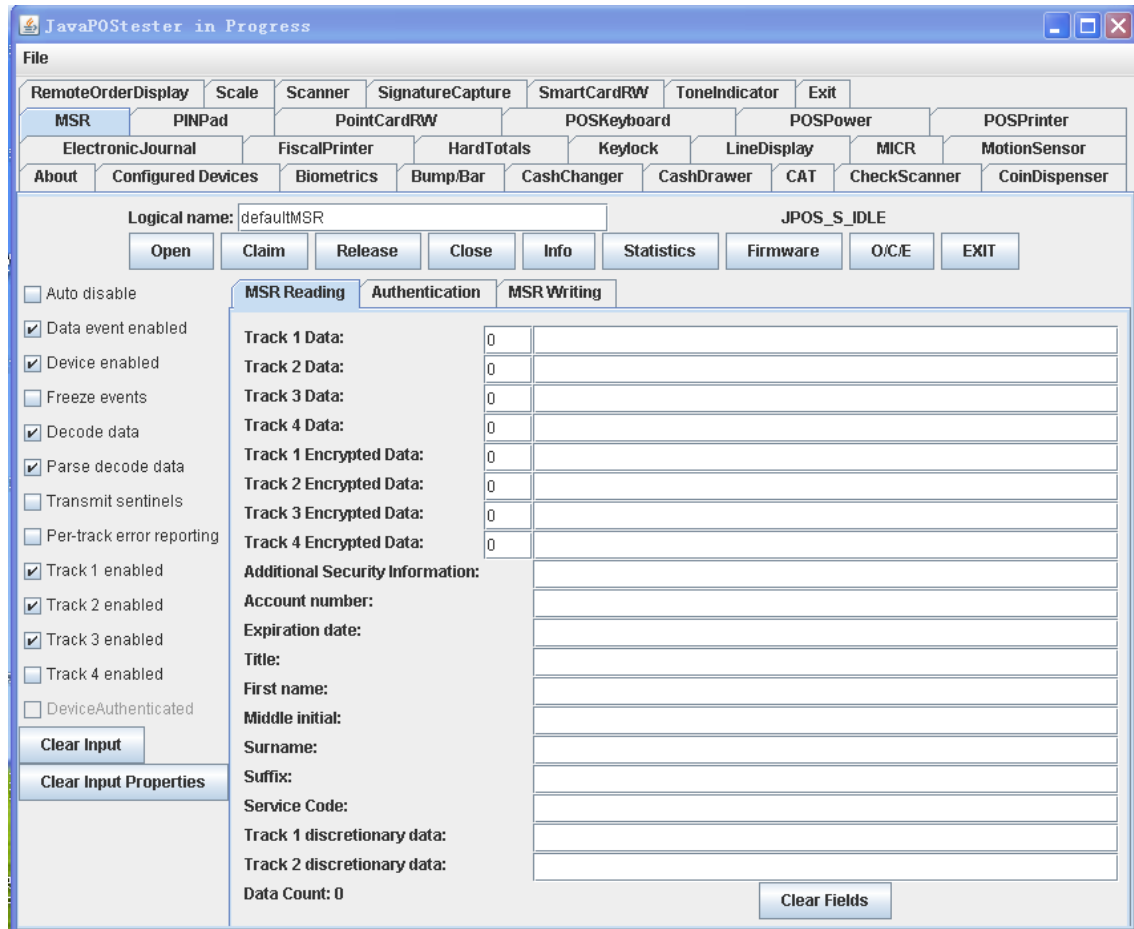
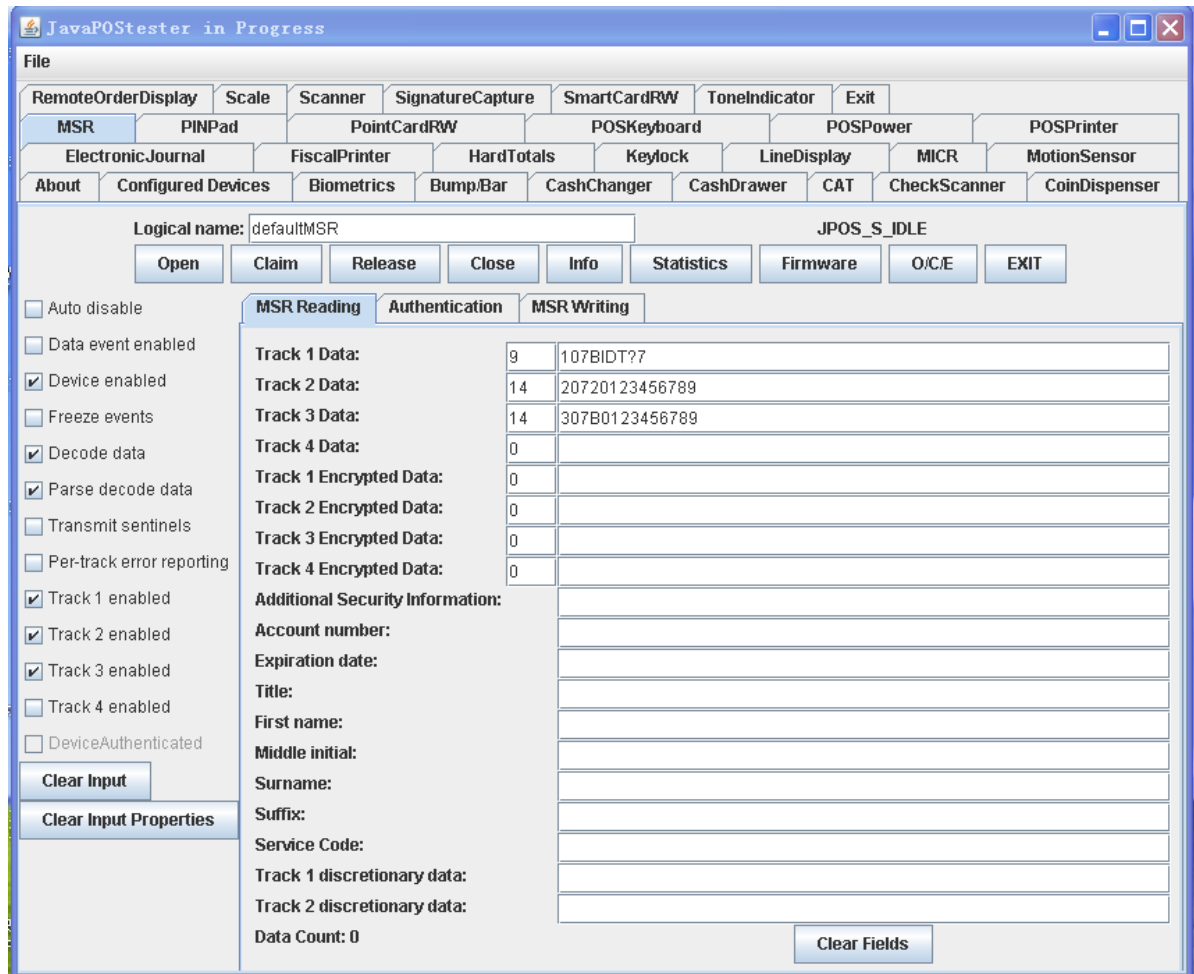4. Click "Open" button, and "Claim", and three "beeps" should sound from the device.

5. Check choices "Data event enabled", "Device enabled"

6. Observe the text field and swipe a card.

7. Click "Release" or "Close" to release or close the JPOS,

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

8. Click "EXIT" to quit the program.

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

6. A Sample for using MSR JPOS

The   following   code   is   the   sample:

```
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){
        MSR msr=new MSR(); /*   new an instance */


        try{
            msr.open(TestConst.testdevicename);
        }catch(JposException e){
            System.err.print(e);
        }
        /**
            Do something here.
         */
        try{
           msr.close();
        }catch(JposException e){
            System.err.print(e);
        }
    }
}
```

Important Note:

If the reader has been authenticated, the application should cancel the authentication before close the POS. Otherwise, the reader will stay in the authentication state 120 seconds.