TRST-A1x JavaPOS Driver Kit

OPERATION MANUAL

This TRST-A1x JavaPOS Driver Kit OPERATION MANUAL (hereinafter referred to as “the GUIDE”) describes the procedures and precautions for using the JavaPOS Driver Kit (hereinafter referred to as “the Kit”).

The GUIDE assumes that the reader is familiar with the following:

- General characteristics of POS peripheral devices
- Java terminology and architecture
- Java for Retail POS (JavaPOS for short) Programmer’s Guide

Notes:

- It is prohibited to use or duplicate a part or whole of the GUIDE without the permission of Toshiba TEC Corporation.
- The GUIDE is subject to change without prior notice.

* Windows and Windows XP are registered trademarks of Microsoft Corporation in the United States and/or other countries.
The official name of Windows is the “Microsoft Windows Operating System”.

* Java and JavaPOS are trademarks of Sun Microsystems in the United States and/or other countries.

Table of Contents

Introduction

The Kit (TRST-A1x JavaPOS Driver Kit) provides the JavaPOS Device Service to be used to develop an application for POS Printer.

Overview of the MANUAL

The GUIDE consists of the following three steps, and explains the demo program enclosed in the Kit. Please follow the steps below:

Step 1. **How to Build An Operating Environment**

This step describes the method to build an environment for using the JavaPOS Device Service.

Step 2. **How to Check Performance of the JavaPOS Device Service**

This step describes the method to test whether or not the test program for checking performance of the JavaPOS Device Service (CheckHealth.jar) successfully runs.

Step 3. **How to Use The JavaPOS Device Service**

This step describes the method to create a unique application using the JavaPOS Device Service.

1. How to Build An Operating Environment

This chapter describes the method to build an environment where the JavaPOS Device Service operates. Please take this step (Step 1) first, then go to Step 2 (Chapter 2 “How to Check Performance of the JavaPOS Device Service”) or Step 3 (Chapter 3 “How to Use the JavaPOS Device Service”).

Supported Product

- TRST-A10-xxx-QM
- TRST-A10-xxx-CN
- TRST-A15-xxx-QM
- TRST-A15-xxx-CN
- TRST-A00

Operating Environment

Performance of the JavaPOS Device Service was checked under the following environment:

Operating system:	Windows 2000, Windows XP, WEPOS, Windows Vista, SUSE Linux Enterprise Desktop 10 SP1, SUSE Linux Enterprise Desktop 11
JavaRuntime:	JRE1.4.2
JavaPOS:	JavaPOS 1.11

Procedure for Building An Operating Environment

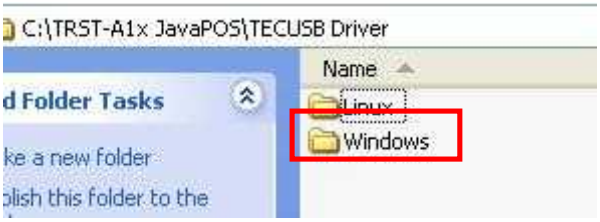
Installation of Java Runtime Environment

Download the file from the following web site and implement the Java Runtime Environment.
http://java.sun.com/products/archive/j2se/1.4.2_16/

Installation of TECUSB

<Windows>

1. Copy of driver module



Copy an “¥TRST-A1x JavaPOS¥TECUSB Driver¥Windows” folder in the suitable place.

2. Execute of Batch file

[Windows 2000 / Windows XP]

Carry out “TECUSB_LIBRARY_SETUP.BAT” in the folder which I stored by procedure 1.

After practice, the following file is copied by a folder of “¥Windows¥system32”.

(in the case of the Windows 2000, ¥Winnt¥system32)

- TECUSB.dll
- LogMngr.dll
- TECUSBJNI.dll
- TECUSBPM.exe

[Windows Vista]
In case of Windows Vista, copy the following files to each directory manual operation.

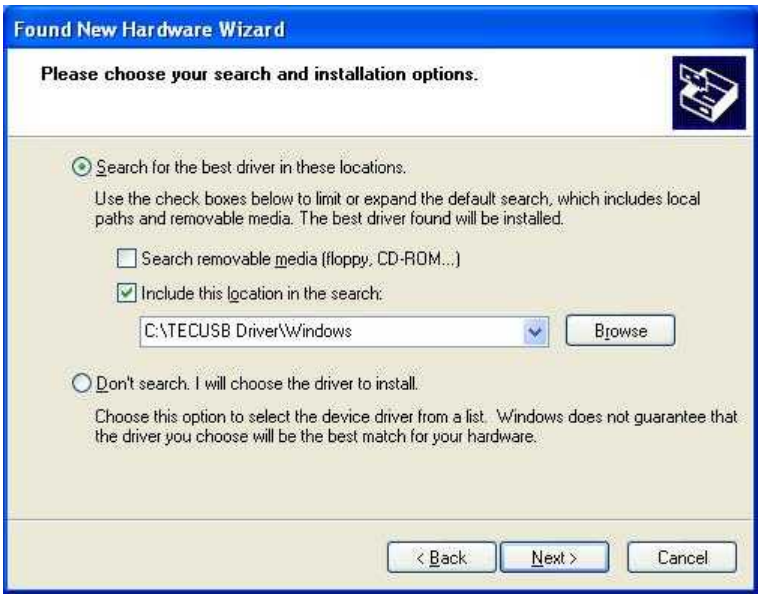
"Windows¥system32" directory

- TECUSB.dll
- LogMngr.dll
- TECUSBJNI.dll
- TECUSBPM.exe

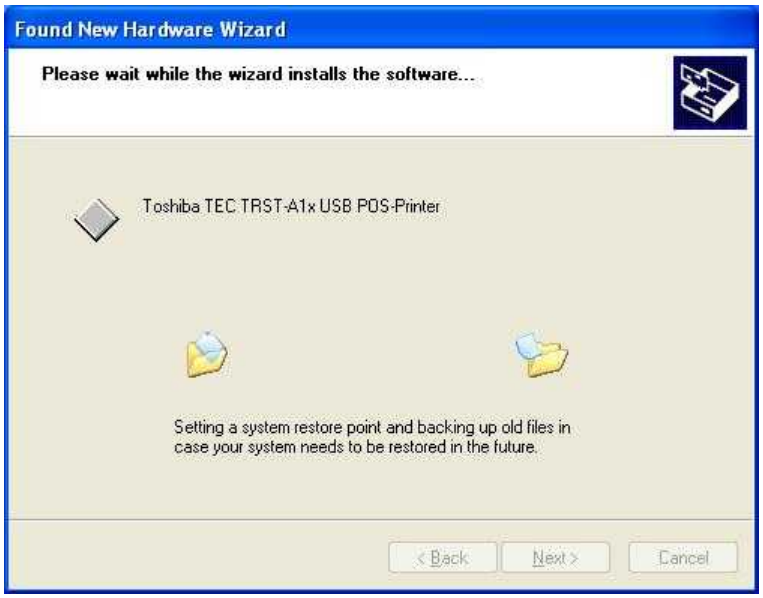
3. Installation of TRST-A1x TECUSB driver
Connect TRST-A1x by USB and turn on a power supply.
The following dialogue is displayed.

Choose "No, not this time". Click on the "Next >" button.

Choose “Install from a list or specific location [Advanced]”. Click on the ”Next >” button.

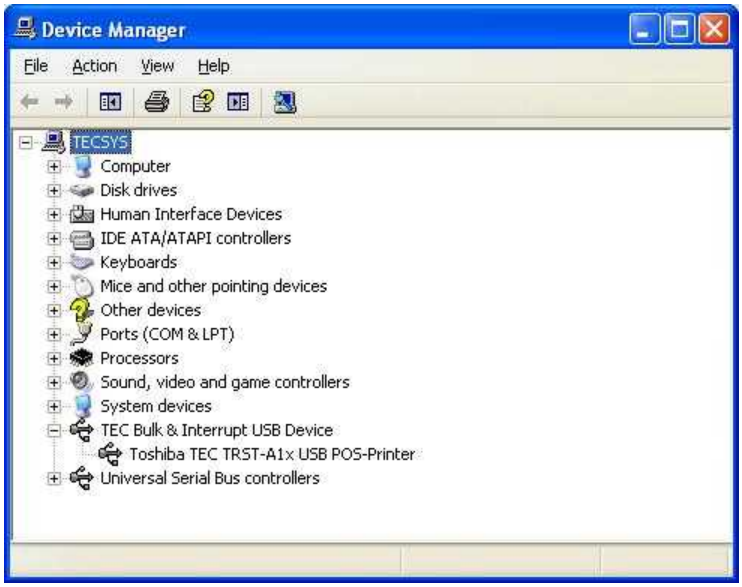


Exclude a check box of “Search removable media [floppy, CD-ROM...]”.
Choose check box of “Include this location in the search.”.
Click on the "Browse" button. And appoint a folder with the "TecUSBDEx.INF" file.
Click on the “Next >” button.



The installation of the TECUSB driver is started.

If an above screen is displayed, it is installation completion.
Click on the ”Finish” button.



Finally start device manager. And confirm that it is installed as above.

< Linux>

1. Copy of driver module

Select “TECUSB Driver” -> “Linux” -> “driver”. Copy the tecusbd.ko to any desired location.
SUSE Linux Enterprise 10 SP1 : use the module in “SLED10SP1” directory.
SUSE Linux Enterprise 11 : use the module in “SLED11” directory.

* tecusbd.ko is a TECUSB driver which runs on Linux.

Select “TECUSB Driver” -> “Linux”. Copy the “libtecusb.so.0.0” to any desired location.

**libtecusb.so.0.0 is a TECUSB library which runs on Linux.

2. Installation of driver

```
]# insmod /home/tec/tecdrv/tecusbd.ko
```

Execute the above commands to install the driver.
(The above is an example when tecusbd.ko has been copied to /home/tec/tecdrv.)

*1 This setup is cleared whenever the operating system is rebooted. This command must be executed every time the operating system is rebooted

3. Confirmation of install

```
]# lsmod
```

Execute the above command to make sure the driver has been successfully installed.

4. Installation of library

```
]# cp -p /home/tec/tecdrv/libtecusb.so.0.0 /usr/lib/
]# ldconfig -n /usr/lib/
]# ln -s /usr/lib/libtecusb.so.0.0 /usr/lib/libtecusb.so
```

Execute the above commands to install the library.
(The above is an example when libtecusb.so.0.0 has been copied to /home/tec/tecdrv.)

2. How to Check Performance of the JavaPOS Device Service

This chapter describes the method to check performance of the JavaPOS Device Service, assuming that the operating environment described in Chapter 1 has been built up.
Here, the setup method is referred to as “PREPARE” and the operation method of the Device Health Check Program as “OPERATION”.

In this chapter, the device health check method is explained for the following devices:
- POSPrinter(TRST-A1x)
- CashDrawer(TRST-A1x)

PREPARE

Copy of Device Health Check Program modules

Open the Kit CD, then copy the Device Health Check Program folder to a desired location in the local computer.

* The subsequent procedures are separately explained for Windows and Linux below.

<Linux>

1. Creation of symbolic link

```
]# ln –s    libTECUSBJNI.so.0.0    libTECUSBJNI.so
```

2. Grant of execute authority

```
]# chmod 775 /home/tec/Device Health Check Program/CheackHealth.sh
```

Execute the above command to grant the CheackHealth.sh file an execute authority.
(The above is an example when the Device Health Check Program folder has been copied to /home/tec.)

Default value

Default value of major parameters is as follows. To change the default value, please refer to the chapter, "3. How to Use the JavaPOS Device Service" in the GUIDE or the setup method in the Application User Manual of each device service.

[TRST-Axx Parallel POS Printer]

[Windows Parallel POSPrinter]
logicalName: WindowsParallelPOSPrinter
portName: LPT1

[Windows Parallel CashDrawer]
logicalName: WindowsParallelCashDrawer
portName: LPT1

[Linux Parallel POSPrinter]
logicalName: LinuxParallelPOSPrinter
portName: /dev/lp0

[Linux Parallel CashDrawer]
logicalName: LinuxParallelCashDrawer
portName: /dev/lp0

[TRST-Axx Serial POS Printer]

[Windows Serial POSPrinter for TRST-A1x]
logicalName: WindowsSerialPOSPrinter
portName : COM1

[Windows Serial CashDrawer]
logicalName: WindowsSerialCashDrawer
portName : COM1

[Linux Serial POSPrinter for TRST-A1x]
logicalName: LinuxSerialPOSPrinter
portName: /dev/ttyS0

[Linux Serial CashDrawer]
logicalName: LinuxSerialCashDrawer
portName: /dev/ttyS0

[Windows Serial POSPrinter for TRST-A00]
logicalName: WindowsSerialPOSPrinter-A00
portName : COM1

[Linux Serial POSPrinter for TRST-A00]
logicalName: LinuxSerialPOSPrinter-A00
portName: /dev/ttyS0

[TRST-A1x USB POS Printer]

[USBPOSPrinter for TRST-A1x-QM model] logicalName: USBPOSPrinter-QM vendorName: 61	[USBCashDrawer for TRST-A1x-QM model] logicalName: USBCashDrawer-QM vendorName: 61
[USBPOSPrinter for TRST-A1x-CN model] logicalName: USBPOSPrinter-CN vendorName: 70	[USBCashDrawer for TRST-A1x-CN model] logicalName: USBCashDrawer-CN vendorName: 70
[USBPOSPrinter for TRST-A00] logicalName: USBPOSPrinter-A00 vendorName: 82	[USBCashDrawer for TRST-A00] logicalName: USBCashDrawer-A00 vendorName: 82

[TRST-A1x LAN POS Printer]

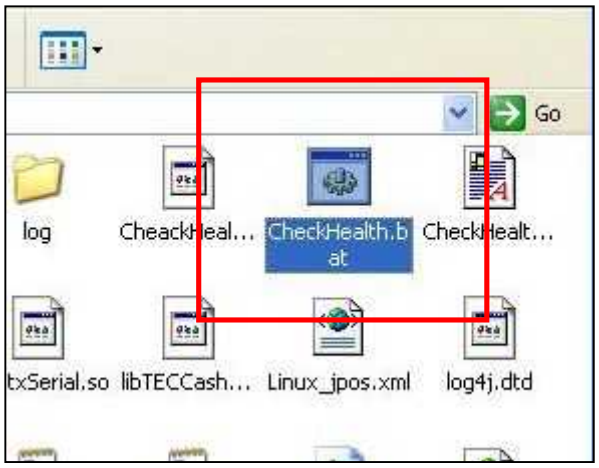
[LANPOSPrinter No.1] logicalName: LANPOSPrinter-1 IPAddress: x.x.x.x	[LANCashDrawer] logicalName: LANCashDrawer IPAddress: x.x.x.x
[LANPOSPrinter No.2] logicalName: LANPOSPrinter-2 IPAddress: x.x.x.x	
[LANPOSPrinter No.3] logicalName: LANPOSPrinter-3 IPAddress: x.x.x.x	

Note : In case of LAN POSPrinter, set IPAddress of TRST-A1x POS Printer before using CheckHealth.

OPERATION

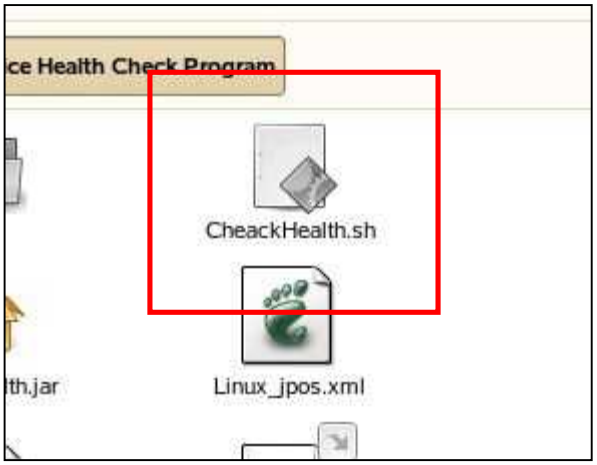
1 Startup of program

<Windows>



Execute the CheckHealth.bat file.

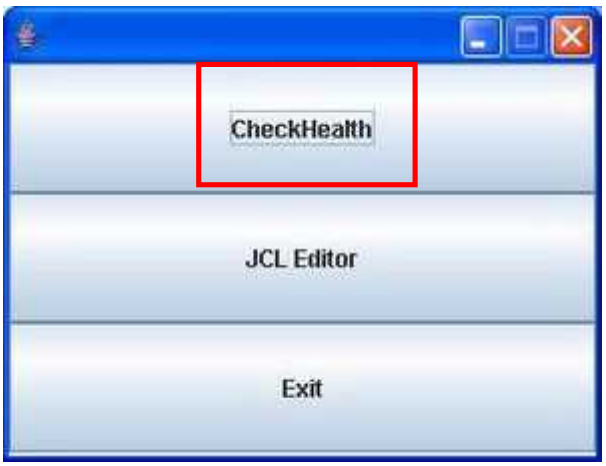
<Linux>



Select Run in Terminal to run CheackHealth.sh.

* The subsequent OPERATION applies both to Windows and Linux.

2 Execution of Device Health Check program



Click on the [CheckHealth] button at the top.

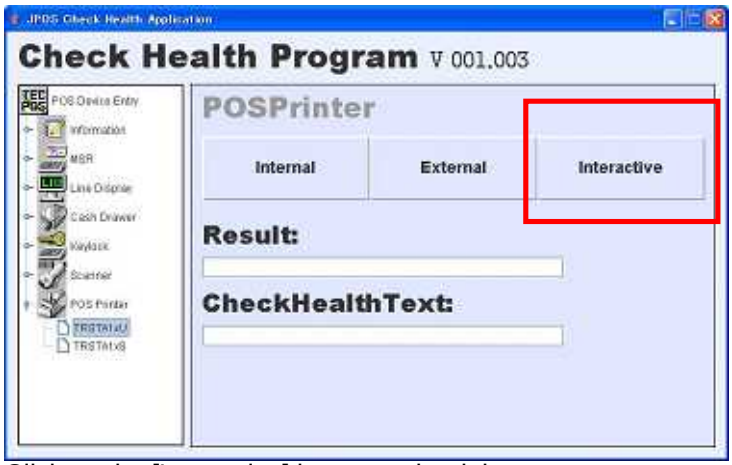
* Note the subsequent OPERATION differs for Keylock, Drawer, MSR, LineDisplay, and Scanner.
* The functions of the JCL Editor are not used this time.

POSPrinter

A-1 POSPrinter panel display

Click on the [TRSTA1xU] node under the [POSPrinter] node in the case of using USBPOSPrinter.
Click on the [TRSTA1xS] node under the [POSPrinter] node in the case of using SerialPOSPrinter.

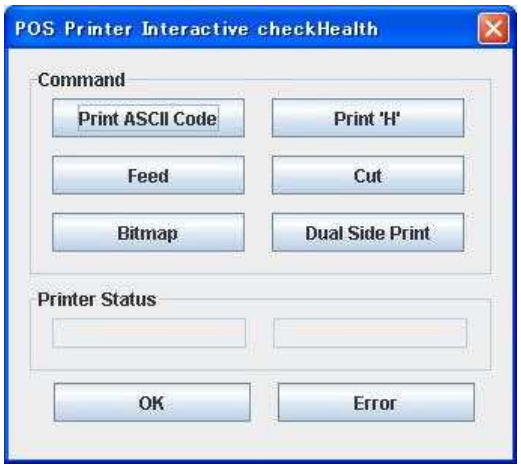
A-2 Call to the Interactive Check Health method



Click on the [Interactive] button at the right.

POSPrinter

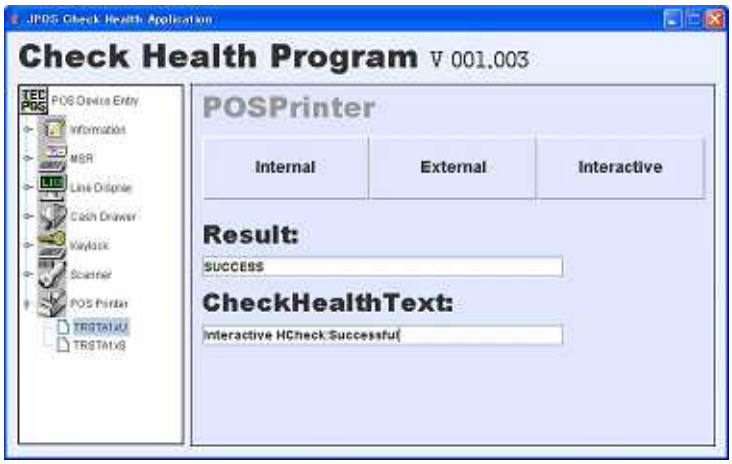
A-3 Execute of Print



When the above window appears, execute each function.

* To exit, click on the [OK] or [NG] button.

A-4 Display of result



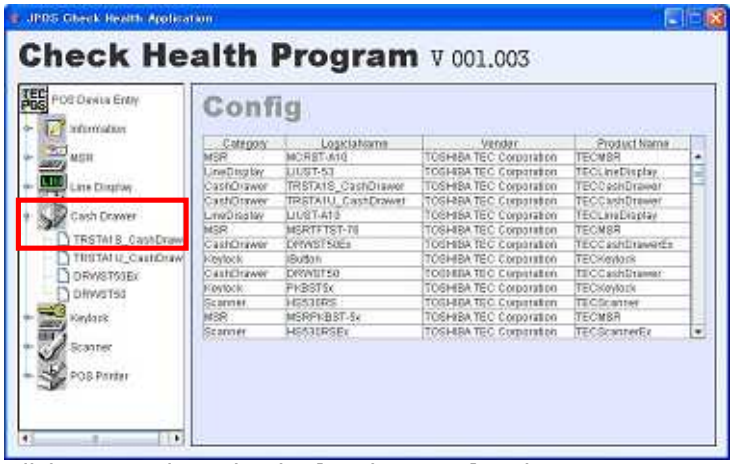
A value is displayed in the [Result] box and the [CheckHealth] Text box.
Either of the following two value combinations will be displayed in these boxes:

- When exited with the [OK] button in Step A-3.
  Result : SUCCESS
  CheckHealthText : Interactive Hcheck:Successful

- When exited with the [NG] button in Step A-3.
  Result : SUCCESS
  CheckHealthText : Interactive Hcheck:Error

CashDrawer

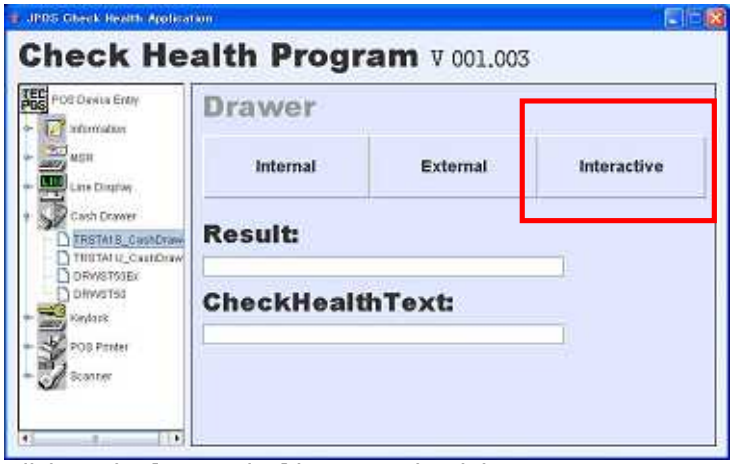
B-1 Drawer panel display



Click on a node under the [CashDrawer] node.

- Cash Drawer connected to TRST-A1x POS Printer

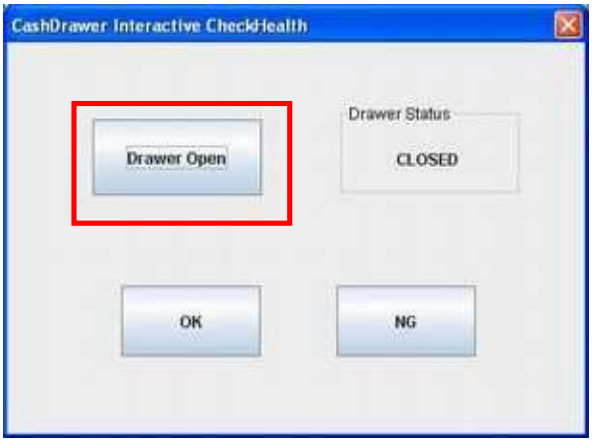
B-2 Call to the Interactive Check Health method



Click on the [Interactive] button at the right.

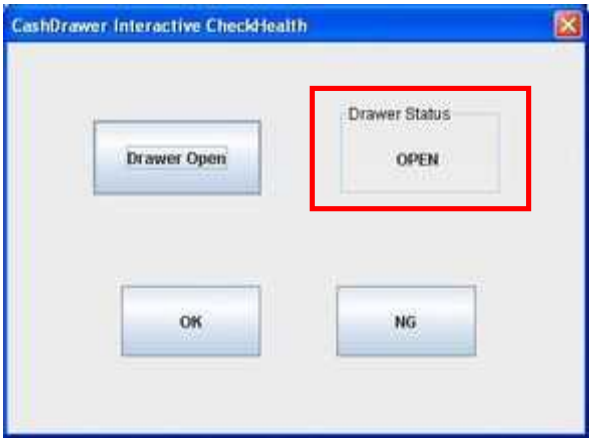
CashDrawer

B-3 Open the drawer



Click on the [Drawer Open] button.

B-4 Get status

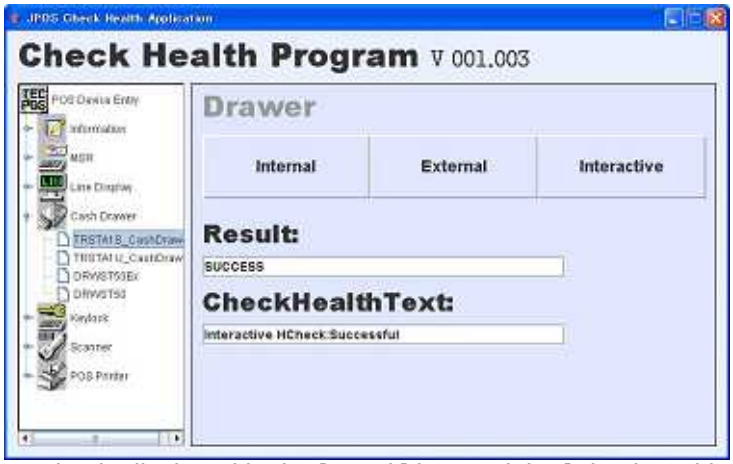


The drawer opens and a message, “OPEN” is displayed on the [Drawer Status] box at the upper right in the window on the screen.

* To exit, click on the [OK] or [NG] button.

CashDrawer

B–5 Display of result



A value is displayed in the [Result] box and the [CheckHealthText] Text box.
Either of the following two value combinations will be displayed in these boxes:

- When exited with the [OK] button in Step B-4.
  Result : SUCCESS
  CheckHealthText : Interactive Hcheck:Successful

- When exited with the [NG] button in Step B-4.
  Result : SUCCESS
  CheckHealthText : Interactive Hcheck:Error

3. How to Use the JavaPOS Device Service

This chapter describes the setups required to use the JavaPOS Device Service, assuming that the operating environment described in Chapter 1 has been built up.

Required files

In order to use the JavaPOS Device Service provided by the Kit, the following files are required besides the library file.

- jpos111.jar
- jpos.xml
- log4j.xml
- log4j.dtd
- log4j-1.2.12.jar
- commons-logging.jar
- RXTXcomm.jar
- swing-layout-1.0.3.jar
- xercesImpl.jar
- xml-apis.jar
- jcl_editor.jar
- rxtxSerial.dll(*1)
- librxtxSerial.so(*2)
- rxtxParallel.dll(*1)
- librxtxParallel.so(*2)
- TECUSB.dll(*3)
- libtecusb.so.0.0(*4)
- TECUSBJNI.dll(*3)
- libTECUSBJNI.so.0.0(*4)
- TECUSBPM.exe(*3)

*1 Required only for Windows
*2 Required only for Linux
*3 Required only for using POSPrinter USB Device Service under Windows
*4 Required only for using POSPrinter USB Device Service under Linux

Description of Files

A destination to save a file may differ depending on the development environment. The following explanation is based on the development using the NetBeans5.5.

Jpos111.jar
(Destination to save): Any location
(Description): JavaPOS Device Control. To be imported when creating an application.
(Available from): JavaPOS-1.11.0-Dist.zip on the web site,
http://www.javapos.com/samplecode.html, or
http://www.javapos.com/index.html

jpos.xml

(Destination to save): Root directory of project

(Description): A device setup file required to operate each Device Service. The following focuses on the major setup items descried in the file. This file is required for operating each Device Service.

*1. Creation of jpos.xml file

An xml file is provided for each device in the JavaPOS folder. When using the xml files, compile all xml files into one file and name it “jpos.xml”.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">

    <JposEntry logicalName="SerialPOSPrinter">
        <creation factoryClass="jpos.toshibatec.posprinter.loader.JavaPOSServiceFactory"
          serviceClass="jpos.toshibatec.posprinter.services.POSPrinterService"/>
        <vendor name="TOSHIBA TEC Corporation" url="http://www.toshibatec.co.jp"/>
        <jpos category="POSPrinter" version="1.11"/>
        <product description="TEC TRST-A1x POS Printer"
          name="TECPOSPrinter" url="http://www.toshibatec.co.jp"/>

        <!--Other non JavaPOS required property (mostly vendor properties and bus specific properties i.e.
RS232 )-->
                        <prop name="deviceBus" type="String" value="RS232"/>
                        <prop name="portName" type="String" value="COM1"/>
                        <prop name="BaudRate" type="String" value="115200"/>
                        <prop name="modelName" type="String" value="TRSTA1x"/>

                        <prop name="country" type="String" value="US"/>
                        <prop name="fontSize" type="String" value="1"/>
                        <prop name="paperWidthMode" type="String" value="0"/>
  </JposEntry>

</JposEntries>
```

The following describes the major setup items. For details of the setup method, please refer to the Application User Manual of each JavaPOS Device Service.

- <JposEntry logicalName="SerialPOSPrinter">
  A description to set a logical device name. Change the shaded area.
- <prop name="portName" type="String" value="COM1"/>
  A description to set COM ports of a device. Change the shaded area.
- <prop name="baudRate" type="String" value="115200"/>
  A description to set baud rate of a device. Change the shaded area.

* 2Defference in descriptions between Windows and Linux
As for portName, "COMX" is used for Windows and "/dev/ttySX" is used for Linux. (X: serial port no.)
Note that COMX starts from 1 while /dev/ttySX starts from 0.

[Windows]
value=”COM1”
value=”COM2”
value=…

[Linux]
value=”dev/ttyS0”
value=”dev/ttyS1”
value=…

log4j.xml

(Destination to save): Root directory of project

(Description): A setup file for a log to be output. To be copied in the directory where the execution file exists. The following focuses on the major setup items descried in the file. Please create your own file.

- <param name="file" value="log/ST-A10.log" />
  A description to set a file name of log to be output.
- <priority value="info" />
  A description to set a log level.

Fatal:	Fatal error	error:	Error
warn:	Warning	info:	Information
debug:	Debug	trace:	Trace

Log4j.dtd

(Destination to save): Root directory of project

(Description): A file to define XML tags. To be copied in the directory where the execution file exists. Please create your own file.

log4j-1.2.12.jar

(Destination to save): Any location

(Description): A library file to output a log. As with JavaPOS DeviceService, this file must be imported in a project.

(Available from): logging-log4j-1.2.12.zip on the web site, http://archive.apache.org/dist/logging/log4j/1.2.12/, or http://logging.apache.org/

commons-logging.jar

(Destination to save): Any location

(Description): A library file to output a log. To be imported when creating an application.

(Available from): commons-logging-1.0.4.zip on the web site, http://archive.apache.org/dist/commons/logging/binaries/, or http://commons.apache.org/logging/

RXTXcomm.jar

(Destination to save): Any location

(Description): A library file to access a Device which uses a COM (component object model). To be imported when creating an application.

(Available from): rxtx-2.1-7-bins-r2.zip on the web site, http://rxtx.qbang.org/pub/rxtx/, or http://users.frii.com/jarvi/rxtx/download.html

swing-layout-1.0.3.jar

(Destination to save): Any location

(Description): A library file to use swing. To be imported when creating an application.

(Available from): swing-layout-1.0.3.jar on the web site, http://java.sun.com/products/archive/jfc/1.0.3/index.html, or http://www.sun.com/

xercesImpl.jar

(Destination to save): Any location

(Description): A library file to convert into text or other XML format. To be imported when creating an application.

(Available from): Xerces-J-bin.2.9.0.zip on the web site, http://apache.adcserver.com.ar/xml/xerces-j/, or http://xerces.apache.org/

xml-apis.jar

(Destination to save): Any location

(Description): A library file to convert into text or other XML format. To be imported when creating an application.

(Available from): Xerces-J-bin.2.9.0.zip on the web site, http://apache.adcserver.com.ar/xml/xerces-j/, or http://xerces.apache.org/

JposEntryEditor.jar

(Destination to save): Any location

(Description): A library file to access an XML file. To be imported when creating an application.

(Available from): jcl2.2.0.zip on the web site, http://Availablefromforge.net/project/showfiles.php?group_id=128804&package_id=141062&release_id=306139, or http://jposloader.Available fromforge.net/downloads/?S=A

JimiProClasses.jar

(Destination to save): Any location

(Description): A library file to access an image file. To be imported when creating an application.

(Available from): jimi1_0.zip on the web site, http://java.sun.com/products/jimi/

rxtxSerial.dll

(Destination to save): Root directory of project

(Description): A library file used to access a serial port under Windows.

(Available from): rxtx-2.1-7-bins-r2.zip on the web site, http://rxtx.qbang.org/pub/rxtx/, or http://users.frii.com/jarvi/rxtx/download.html

librxtxSerial.so

(Destination to save): Root directory of project

(Description): A library file used to access a serial port under Linux.

(Available from): rxtx-2.1-7-bins-r2.zip on the web site, http://rxtx.qbang.org/pub/rxtx/, or http://users.frii.com/jarvi/rxtx/download.html

rxtxParallel.dll

(Destination to save): Root directory of project
(Description): A library file used to access a parallel port under Windows.
This is a RXTX parallel library customized by TTEC.
It is based on rxtx-2.1-7(LGPL). “rxtxSerial.dll” file is necessary to use this library.
(Available from): This Kit. Click on “TEC RXTX Parallel Library Source”.
It is bundled with rxtxParallel.zip.

librxtxParallel.so

(Destination to save): Root directory of project
(Description): A library file used to access a parallel port under Linux.
This is a RXTX parallel library customized by TTEC.
It is based on rxtx-2.1-7(LGPL). “librxtxSerial.so” file is necessary to use this library.
(Available from): This Kit. Click on “TEC RXTX Parallel Library Source”.
It is bundled with rxtxParallel.zip.

TECUSB.dll

(Destination to save): C:¥Windows¥system32 or C:¥WINNT¥system32
(Description): Library of TECUSB driver for Windows.
(Available from): This Kit. Click on “TECUSB Driver”, then “Windows”.

LogMngr.dll

(Destination to save): C:¥Windows¥system32 or C:¥WINNT¥system32
(Description): Library of TECUSB driver for Windows.
(Available from): This Kit. Click on “TECUSB Driver”, then “Windows”.

libtecusb.so.0.0

(Destination to save): /usr/lib directory
(Description): Library of TECUSB driver for Linux.
Regist it with shared libraries and make a link file called libtecusb.
Ex)]# cp -p libtecusb.so.0.0 /use/lib/
]# ldconfig –n /usr/lib/
]# ln –s /usr/lib/libtecusb.so.0.0 /usr/lib/libtecusb.so
(Available from): This Kit. Click on “TECUSB Driver”, then “Linux”.

TECUSBPM.exe

(Destination to save): C:¥Windows¥system32 or C:¥WINNT¥system32
(Description): An USB power management process for Windows.
It is nesessary to use TRST-A1x-U on Windows
(Available from): This Kit. Click on “TECUSB Driver”, then “Windows”.

TECUSBJNI.dll

(Destination to save): C:¥Windows¥system32 or C:¥WINNT¥system32
(Description): An application programming interface (API) to be used to access the Windows TECUSB driver from Java.
(Available from): This Kit. Click on “TECUSB Driver”, then “Windows”.

libTECUSBJNI.so.0.0

(Destination to save): Root directory of project
(Description): An application programming interface (API) to be used to access the Linux.
Make a link file called libTECUSBJNI.so and use it.
Ex)]# ln –s libTECUSBJNI.so.0.0 libTECUSBJNI.so
(Available from): This Kit. Click on “TECUSB Driver”, then “Linux”.