EPSON

EPSON JavaPOS ADK

Users Reference

Version 1.52L Apr. 2002

Cautions

1. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SEIKO EPSON Corporation.
2. The contents of this document are subject to change without notice. Please contact us for the latest information.
3. While every precaution has been taken in the preparation of this document, SEIKO EPSON Corporation assumes no responsibility for errors or omissions.
4. Neither is any liability assumed for damages resulting from the use of the information contained herein.
5. Neither SEIKO EPSON Corporation nor its affiliates shall be liable to the purchaser of this product or third parties for damages, losses, costs, or expenses incurred by the purchases or third parties as a result of: accident, misuse, or abuse of this product or unauthorized modifications, repairs, or alterations to this product, or (excluding the U.S.) failure to strictly comply with SEIKO EPSON Corporation's operating and maintenance instructions.

Trademarks

Linux is a trademark or registered trademark of Linus Torvalds in the U.S.A. and/or other countries.

Red Hat is a registered trademark of Red Hat, Inc.

Contents

Chapter 1 Introduction

This manual gives an overview of EPSON JavaPOS ADK 1.52L, and describes its installation and setup. Refer to "Java for Retail POS Programming Guide Version 1.5" prepared by Java for Retail POS Committee for details of the API functions and application.
In addition, refer to "EPSON CheckScanner Programming Guide version 1.5." to use the CheckScanner device.

1.1 Terms described in this manual

- "Java for Retail POS Programming Guide version 1.5" is abbreviated as "JavaPOS PG".
- "EPSON CheckScanner Programming Guide version 1.5" is included in "Java for Retail POS Programming Guide version 1.5" or "JavaPOS PG".
- Configuration tool for EPSON JavaPOS ADK is abbreviated as "EPSON JavaPOS Entry Editor".
- "EPSON JavaPOS ADK" indicates "EPSON JavaPOS ADK 1.52L".
- "Java2 SDK" indicates "Java2 SDK, Standard Edition Version1.3.1_02 for Linux/x86".

Chapter 2 Overview

This chapter gives an overview of EPSON JavaPOS ADK.

2.1 Characteristics

EPSON JavaPOS ADK is driver software that is compliant with the JavaPOS standard specification. EPSON JavaPOS ADK supports the Linux environment as its target platform.

Target devices are EPSON POS printers, MICRs, line displays, cash drawers, barcode scanners (for the Japanese market), magnetic stripe reader, keylock, and CheckScanner. Use the JavaPOS APIs by referring to JavaPOS PG so that you can create programs that are independent of devices but make use of the devices' capabilities.

Common Java environments are used for creating applications. This allows you to develop applications in your familiar development environment.

The applications you develop can be executed on a POS terminal or a personal computer running Red Hat's Linux (for information on versions that can be used, see "2.2 Operating environments"). Before executing the applications, be sure to connect the required devices properly, and describe the devices' connection information in EPSON JavaPOS's configuration file.

2.2 Operating environments

EPSON JavaPOS ADK operates in the following environments.

2.2.1 Host computer

IBM PC/AT compatible

The following specification is recommended.

CPU:	Celeron 366 MHz or higher
Memory:	128 MB or more
Hard Disk:	100 MB or more available disk

2.2.2 Operating system

Red Hat 6.2 English (kernel 2.2.14)

Red Hat 7.1 (kernel 2.4.2)

2.2.3 Java environments

Use the following Sun Microsystems Java as the operating environments.

-Java2 SDK, Standard Edition Version1.3.1_02 for Linux/x86

The following CommAPI is required when you use serial interface devices.

-RxTx(serial and parallel port API) 1.4-9

For the CheckScanner device, we recommend using the "Java Advanced Imaging" package from Sun Microsystems.

- Java Advanced Imaging 1.1.1 for Linux

All of the above software are provided free of charge. See "3.1 Obtaining the required package" for more information.

EPSON’s USB driver is required when using a device with the USB interface. EPSON’s USB driver is contained in EPSON JavaPOS ADK. The kernel version that this driver supports currently is as follows:

- 2.4.2-2 (Red Hat 7.1)

Chapter 3 Installation

3.1 Obtaining the required package

EPSON JavaPOS ADK provides the following files:

- EPSON_JavaPOS152_Install.tar.gz (EPSON JavaPOS ADK itself)
- EJEE2.0.1_Install.tar.gz(EPSON JavaPOS Entry Editor)
- install

Additional packages for using EPSON JavaPOS ADK as well as information on how they can be obtained are listed as follows:

- Java2 SDK, Standard Edition Version1.3.1_02 for Linux/x86
  It can be downloaded free of charge from the following URL address.
  http://java.sun.com/
- Java Communications API 2.0(for Solaris/x86)
  It can be downloaded free of charge from the following URL address.
  http://java.sun.com/products/javacomm/
- Java Advanced Imaging 1.1.1 for Linux
  It can be downloaded free of charge from the following URL address.
  http://java.sun.com/products/javacomm/
- RxTx(serial and parallel port API)1.4-9
  It can be downloaded free of charge from the following URL address.
  http://www.rxtx.org/
- Xerces(XML parser)1.4.3
  It is included in the EPSON JavaPOS ADK package and does not require installation. It can also be downloaded free of charge from the following URL address.
  http://xml.apache.org/xerces-j/

3.2 Installing over previous version

This new version installer creates the "javapos" and "ejee" sub-directories under the specified install directory; EPSON JavaPOS ADK and EJEE utility are installed in these directories. If EPSON JavaPOS ADK of a previous version has been installed, delete the "javapos" and the "ejee" directories or change their names before the new installation.

Be sure to modify the class path.

3.3 Installation overview

This section describes an installation procedure of EPSON JavaPOS ADK, as well as the package required for using EPSON JavaPOS ADK.

The following shows a general procedure to use when installing EPSON JavaPOS ADK for the Linux environment. See "3.4 Installation procedure" for details on the procedure.

1. Install Java2 SDK
2. Install Java Communications API and RxTx(Communications Port API)
3. Install EPSON JavaPOS ADK and EJEE utility

If EPSON IM series is used, you need to change the serial port setting when using the COM3/4 serial port. See "4.1.1 Configuring COM3/4 port" for details on the procedure.

If a magnetic stripe reader or keylock is used on the IM series, you need to use the EPSON keyboard hook driver. See "4.1.2 Using the EPSON keyboard hook driver" for details on the procedure.

The installation requires that you log on using the "root" account or an account that has the "root" authority. If you log on using an account other than "root", the installation cannot be performed correctly.

3.4 Installation procedure

This section describes how to install EPSON JavaPOS ADK and the required packages for using EPSON JavaPOS ADK. In the explanation, the bold portion indicates that you need to input the required information.

3.4.1 Installation of Java2 SDK

The following uses Java2 SDK as an example to illustrate the installation procedure Version1.3.1_02 is used, which is available in a binary self-extracting format. For installation of other formats, read the information available on the same URL address from which you have downloaded Java2 SDK.

Step1: Installing Java2 SDK

Copy j2sdk-1_3_1_02-linux-i386.bin to the /usr/local directory.

Example:

cp j2sdk-1_3_1_02-linux-i386.bin /usr/local/

Move to the /usr/local directory and execute j2sdk-1_3_1_02-linux-i386.bin. When the execution is completed, a directory named /usr/local/jdk1.3.1_02 is created and Java2 SDK is installed in this directory.

Example:

cd /usr/local/
chmod +x j2sdk-1_3_1_02-linux-i386.bin
./j2sdk-1_3_1_02-linux-i386.bin

Step2: Setting up the environment variables

You then need to set up the environment variables. There are many ways you can set them up, and the following is an example.

Move to the /root directory.

Example:

cd /root

Add the following environment variables to the file.bash profile.

```
JDK_HOME=/usr/local/jdk1.3.1_02
PATH=$PATH:$JDK_HOME/bin

export PATH
```

To make the environment variables effective, log off, and then log on again.

Step3: Checking the installation

Make sure JDK is correctly installed. From any directory, use the following command line, for example, to check the version of Java.

Example:

```
# java –version
  java version "1.3.1_02"
  Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_02-b02)
  Java HotSpot(TM) Client VM (build 1.3.1_02-b02, mixed mode)
```

Step4: Installing Java Advanced Imaging 1.1.1 for Linux (Required only when using the CheckScanner device)

Copy jai-1_1_1-lib-linux-jre.bin to the /usr/local/jdk1.3.1_02/jre directory.

Example:

```
# cp jai-1_1_1-lib-linux-jre.bin /usr/local//jdk1.3.1_02/jre
```

Move to the /usr/local/jdk1.3.1_02/jre directory and execute jai-1_1_1-lib-linux-jre.bin.

Example:

```
# cd /usr/local/jdk1.3.1_02/jre
# chmod +x jai-1_1_1-lib-linux-jre.bin
# ./jai-1_1_1-lib-linux-jre.bin
```

3.4.2 Installing RxTx

3.4.2.1 Before using RxTx

RxTx is a communication library. (Its license follows LGPL. For information on the license, refer to the license agreement included in the RxTx package.) If the COM3/4 port (for TM-T88IIR, etc.) is used, you need to modify the serial driver to match the IRQ, as described in "4.1.1 Configuring COM3/4 port".

When you run a JavaPOS application, the time to open a serial port can be shortened by adding options. See "3.4.2.3 How to shorten the time to open a serial port" for details.

When using the CheckScanner device, to improve the reading speed of image data, we recommend to modify RxTx before using it. Refer to the CheckScanner manual on how to modify RxTx.

3.4.2.2 Installation

Note that the installation requires Java2 SDK and Java Communications API 2.0(for Solaris/x86). For environments where only Java2 Runtime Environment is installed, such installation is not possible.

Step1: Installing Java Communications API 2.0(for Solaris/x86)

Copy javacomm20-x86.tar.Z to the /usr/local directory.

Example:

cp javacomm20-x86.tar.Z /usr/local/

Move to the /usr/local directory, and then expand javacomm20-x86.tar.Z

Example:

cd /usr/local

tar –xvzf javacomm20-x86.tar.Z

From Java Communications API for Solaris/x86 (i.e. files created when javacomm20-x86.tar.Z is expanded), copy the necessary files to the JDK directory.

Example:

 # cd /usr/local/commapi

cp comm.jar /usr/local/jdk1.3.1_02/jre/lib/ext/

cp javax.comm.properties /usr/local/jdk1.3.1_02/jre/lib/

Step2: Expanding RxTx

Copy rxtx-1.4-9.tar.gz to the /usr/local directory.

Example:

cp rxtx-1.4-9.tar.gz /usr/local/

Move to the /usr/local directory, and then expand rxtx-1.4-9.tar.gz.

Example:

cd /usr/local

tar –xvzf rxtx-1.4-9.tar.gz

Step3: Installing RxTx

Install RxTx.

Example:

cd /usr/local/rxtx-1.4-9

./configure

make install

Step4: Checking the installation of RxTx

Run BlackBox included with Java Communications API (for Solaris/x86) to make sure RxTx is installed correctly.

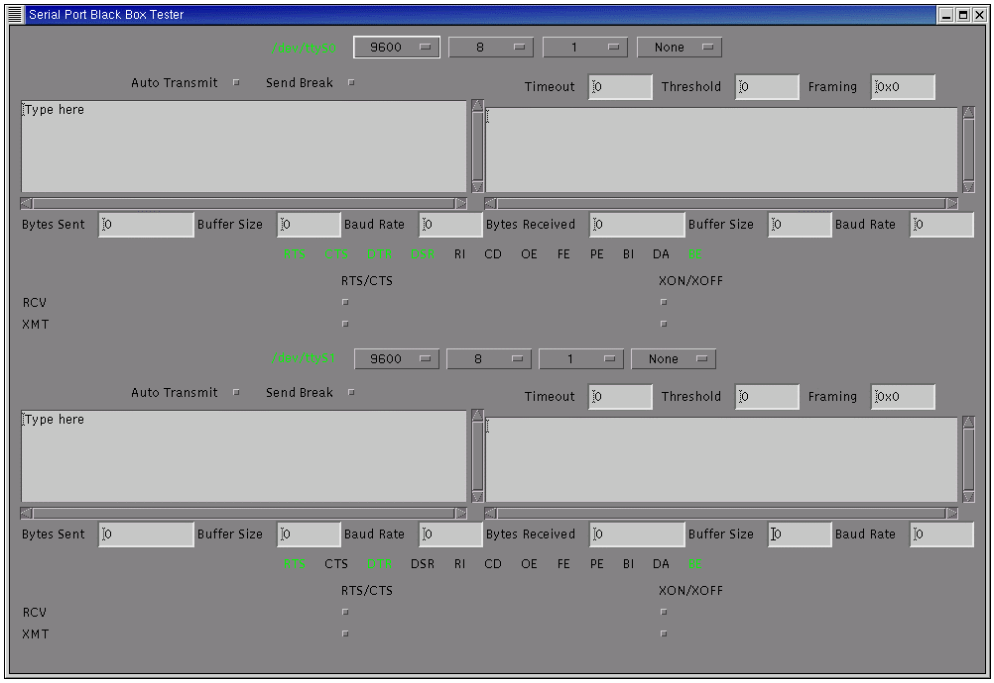
Example:

cd /usr/local/commapi/samples/BlackBox

java –classpath ./BlackBox.jar BlackBox

When BlackBox starts, the screen shown below appears. After setting the communications parameters correctly, enter some text in the text box with "Type here" displayed. When you enter some text, the text data is output to the

appropriate COM port. For details, refer to Readme of BlackBox.



3.4.2.3 How to shorten the time to open a serial port

By default, RxTx is configured to search files under the /dev directory to look for a serial port that can be read from and written to. It takes time when the port is opened for the first time.

By using the "-Dgnu.io.rxtx.SerialPorts" option to specify your desired serial port, the time to open the port when you run Java can be shortened. If you want to specify more ports, separate their names with ":". For example, if you use COM1 and COM2 only, run Java with the following option.

java -Dgnu.io.rxtx.SerialPorts=/dev/ttyS0:/dev/ttyS1 MyApp

Refer to the "INSTALL" file included in the RxTx package for more information.

3.4.3 Installing EPSON JavaPOS ADK and EJEE utility

This section gives an installation overview of EPSON JavaPOS ADK and EJEE utility, as well as their special notes.

3.4.3.1 Overview of installer

The installer is included in EPSON JavaPOS ADK. The installer performs the following tasks.

- Install EPSON JavaPOS ADK and the EJEE utility
- Install the USB driver for TM
- Install the keyboard hook driver
- Install epson.trace.properties(property file for trace)
- Install device controls/services for the development environment

3.4.3.2 Starting installer

Store the following files in the same directory.

- install
- EPSON_JavaPOS152_Install.tar.gz
- EJEE2.0.1_Install.tar.gz

Move to this directory and then execute the following command to start the installer.

sh install

3.4.3.3 Installation procedure

The installation is interactive. During the installation, messages (indicated with "[ ]") appear. The following describes the messages and their special notes.

Step1: Specifying a directory to expand package

When the installer starts, the following message appears. Specify a directory to expand EPSON JavaPOS ADK and the EJEE utility.

[EPSON JavaPOS ADK 1.52L is about to be installed.
Specify the directory to expand this software.
Directories javapos and ejee are made under the specified directory. The necessary files are expanded in the directories.
Specify a location to expand EPSON JavaPOS ADK 1.52L:]

For example, if you specify “/usr/local” as the directory to expand, the following directories are made and EPSON JavaPOS ADK is installed.

- /usr/local/javapos (EPSON JavaPOS ADK install directory)
- /usr/local/ejee (EJEE utility install directory)

Step2: Specifying a path to jpos.xml to read at when JavaPOS starts

After expanding the package, configure jpos.properties. The following message appears.

[If you use EPSON JavaPOS ADK 1.52L, jpos.xml made by EJEE utility should be read when JavaPOS is used.
Specify a path to jpos.xml to read when JavaPOS starts. When you push the return key without entering a path, jpos.xml in the directory that the application starts in is set to be read. [Default]

Note:
When you specify a relative path, use a path from the directory where the application starts.
If jpos.xml is not read, you cannot use EPSON JavaPOS ADK.

Specify the directory that jpos.xml locates:]

This step modifies "jpos.properties". The location of the xml files used in EPSON JavaPOS ADK is written in this file. When you run JavaPOS, copy or link the jpos.xml file that EJEE utility has created, to the directory specified here.

When you install JavaPOS, there is no problem if the specified directory does not exist.

Step3: Specifying the directory that the device control and device service are installed

This step installs the device control and device service in your development environment. In addition, other necessary libraries are also installed at the same time. The following message appears.

[Device Controls and Device Services of EPSON JavaPOS ADK 1.52L are about to installed.
Specify the directory to install:]

Specify the directory for development environment. If the specified directory does not exist, the following message appears, asking if you want to make a directory.

The following files are copied in the specified directory.

- epsonJposControl15.jar
- epsonJposService152.jar
- epsonJposServiceCommon.jar
- xerces.jar

If you use EPSON JavaPOS ADK, these three files must be added to the class path.

Step4: Installing the USB driver for TM

If the version of the kernel is 2.4 or later, choose whether to install the USB driver for TM. The following message appears.

[Do you want to install USB driver for TM? [Y/N]:]

The USB driver for TM is needed for devices using the USB interface. Input [Y] if you want to install it, or [N] otherwise.
If you choose the USB driver for TM, a command to load the driver is added to /etc/rc.modules. When the kernel starts, the driver is also loaded.

Step5: Installing keyboard hook driver

Use this step to choose whether to install the keyboard driver. The following message appears.

[Do you want to install keyboard hook driver? [Y/N]:]

The keyboard hook driver is needed to use MSR and keylock. Input [Y] if you want to install it, or [N] otherwise.

Step6: Installing property files for trace

As the final step, copy the property file "epson.trace.properties" for trace of EPSON JavaPOS ADK to the Java2 SDK environment. Copy it to the following directory.

<directory where Java2 SDK is installed>/jre/lib

If the file epson.trace.proerties exists already, the following message appears, asking whether you want to update the file.

[The directory already contains a file that has same name as the property file for trace called epson.trace.proerties.
Do you want to replace the existing file with new one? [Y/N]:]

If you want to maintain the previous trace configuration, input [N].

Step7: Completing installation

If the installation is completed correctly, the following message appears.

[EPSON JavaPOS ADK 1.52L has been installed.]

3.4.3.4 Error messages and troubleshooting

If an error occurs during the installation, the following error message appears. Use the solutions described to solve the problems.

[Message 1]

[The specified directory does not exist. Do you want to make the directory? [Y/N]:]

This message indicates the specified directory does not exist.

[Solution 1]

If you want to make the directory input [Y] or [y]. If you want to finish the installation without making the directory, input [N] or [n].

[Message 2]

[Expansion of [*file name*] is failed.]

This message indicates expansion of the compressed file (for EPSON JavaPOS ADK and EJEE utility) has failed.

[Solution 2]

Check the following and run the installer again.

- Did you run the installer with the administrator account?
- Do all files described in "3.4.3.2 Starting installer" exist in the same directory?
- Did you run the installer in the directory where the installer exists?

[Message3]

[Acquirement of Java home directory and Java vender name is failed.]

This message appears when the installer cannot get information about Java2 SDK .

[Solution 3]

Check if Java2 SDK is installed correctly.

[Message 4]

[The vender of installed JDK is not supported.]

[Solution 4]

Check if EPSON JavaPOS ADK supports the installed Java2 SDK.

[Message 5]

[Making directory [*directory name*] is failed.]

[Solution 5]

Check if the current account has the administrator authority.

[Message 6]

[Installation of keyboard hook driver is failed.]

[Solution 6]

Check if the current account has the administrator authority.

[Message 7]

[Installation of property file for trace is failed.]

[Solution 7]

Check if the current account has the administrator authority.

[Message 8]

[Installation of Device Controls and Device Services is failed.]

[Solution 8]

Check if the current account has the administrator authority.

[Message 9]

[The USB driver for TM could not be installed.]

[Solution 9]

Check if the current account has the Administrator rights.

In addition, make sure the version of the kernel in use matches the version of the kernel indicated by "2.2.3 Java environments".

3.5 Uninstallation

3.5.1 Uninstallation procedure(common)

This chapter describes the procedure to use to uninstall EPSON JavaPOS ADK. To uninstall EPSON JavaPOS ADK, delete the directories and files created during the installation.

Step1: Uninstalling EPSON JavaPOS ADK

Delete the directory used in "Step1: Specifying the directory to expand" of "3.4.3.3 Installation procedure".

For example, if you specified the "/usr/local" as the directory to expand files, delete the directories following the example procedure described below.

(The following example deletes all contents in the directory. If there are some files you want to keep, copy them to a different directory before removing the directory.)

Example:

rm –rf /usr/local/javapos

rm –rf /usr/local/ejee

Step2: Delete device controls and device services

Delete the device controls and device services set up in "Step3: Specifying the directory that the device control and device service are installed" of "3.4.3.3 Installation procedure".

For example, if you specified "/usr/local/MyApp" as the directory to install, delete the files following the example procedure described below.

Example:

cd /usr/local/MyApp

rm –f epsonJposControl15.jar

rm –f epsonJposServiceCommon.jar

rm –f epsonJposService152.jar

rm –f xerces.jar

Step3: Deleting the USB driver for TM

Delete the USB driver for TM set up in "Step4: Installing USB driver for TM" of "3.4.3.3 Installation procedure".

First, unload the driver if it is loaded. Then, turn off all printers connected with USB. Use the procedure described in the following example to unload the driver.

Example:

 # /sbin/rmmod usbtm

Delete the files using the following procedure (example).

Example:

 # cd <the directry Java2 SDK is installed>/jre/lib/i386
 # rm –f libusbtm_jni.so
 # cd /lib/modules/misc
 # rm –f usbtm.o
 # cd /dev/usb
 # rm –f usbtm*

If you have chosen [Y] in "Step4: Installing USB driver for TM" of "3.4.3.3 Installation procedure", use an editor to open /etc/rc.modules and delete the line containing "/sbin/insmod usbtm".

Step4: Delete the property files for trace

Delete the property files for tracing set up in "Step7: Installing the property files for tracing" of "3.4.3.3 Installation procedure

Delete the files following the example procedure described below.

Example:

cd <*directory where Java2 SDK is installed*>/jre/lib
rm –f epson.trace.properties

3.5.2 Uninstallation procedure(for IM-310/IM-600 only)

To uninstall, stop the keyboard hook driver first. For example, if EPSON JavaPOS ADK is installed in "/usr/local/javapos", follow the example procedure described below.

Example:
cd /usr/local/javapos
./k_unload -u

Next, delete the installed keyboard hook driver.

Example:
cd /usr/lib
rm –f kbh.o
 # rm –f libJposKbdInterface.so
cd <*directory where Java2 SDK is installed*>/jre/lib/i386
rm –f kbh.o
 # rm –f libJposKbdInterface.so

As the final step, delete EPSON JavaPOS ADK. See "3.5.1 Uninstallation procedure(common " for details.

Chapter 4 Configuration

This chapter describes the configuration of each device.

4.1 Configuring software

4.1.1 Configuring COM3/4 port

If the COM3/4 port is used as the serial port, you need to modify the IRQ of the serial driver, as shown in the following example.

Move to the /root directory.

Example:
cd /root

Add commands to ".bash_profile" to modify the IRQ of the serial driver. For example, to change the IRQ of COM3(/dev/ttyS2) to 11, add the following command.

```
/bin/setserial /dev/ttyS2 irq 11
```

Log off and log in again to make the added commands effective.

Set IRQ to 11 for COM3(/dev/ttyS2), 10 for COM4(/dev/ttyS3) on EPSON IM series.

4.1.2 Using the EPSON keyboard hook driver

Include the EPSON keyboard hook driver when you use the magnetic stripe driver and keylock on EPSON IM series.

4.1.2.1 Starting and stopping the EPSON keyboard hook driver

Make sure that "kbh_load" and "k_unload" exist in the directory that EPSON JavaPOS ADK installed (such as /usr/local/javapos). (If they do not exist, you

may have selected [N] when the following message appears during the installation:[Do you want to install keyboard hook driver?[Y/N]:]. Check "3.4.3.3 Installation procedure" and then install the EPSON JavaPOS ADK again.
"kbh_load" starts the keyboard hook driver and hooking process.
"k_unload" stops the hooking process and restarts.

(1) Starting the keyboard hook driver
Run "kbh_load". The keyboard hook driver and hook process starts automatically. Be sure to run the driver before using the magnetic stripe reader or the keylock service with JavaPOS.

If you want to make the keyboard hook driver and the hook driver start automatically when the OS starts, add the process that runs "kbh_load" to the process of starting the OS. The following is an example.

Edit or create ".bash_profile" under the "/root". Add the following command to the file.

<directory where EPSON JavaPOS ADK is installed>/ kbh_load
(Example: /usr/local/javapos/kbh_load)

Log off and then log in again to make the added command effective.

(2) Stopping and restarting the hooking process
Run "k_unload -u". The hooking process stops.
Run "k_unload -s" to restart the hooking process that has stopped.

4.1.2.2 Special notes

(1) Position of the keylock after the hook driver starts

Since the keylock of EPSON IM series performs one-way communication from devices to the computer, the key should be used (the device sends data to the computer) at least once to check the keylock position. After the hook driver starts, the keylock position is unknown until the first key operation. In this case, if you start the keylock service of JavaPOS to check the position of keylock, the value (LOCK_KP_UNKNOWN) returned indicates the position is unknown. To avoid this situation, be sure to change the keylock position at least once, before starting the keylock service after the hook driver has started. (For example, when the keylock service starts on the JavaPOS application, display a message to ask the user to change the keylock position.)

The keylock position is stored in the hook driver. Therefore, once the keylock operation has been performed, the value that indicates the unknown position does not return. You can still know the current keylock position even when keylock restarts.

(2) The hooking process of keyboard hook driver

Magnetic stripe reader and keylock that can be installed on the EPSON IM series send data to the computer as keyboard data. The keyboard hook driver watches the input keyboard data and separates data to keyboard, magnetic stripe reader and keylock. The separated data of keyboard and magnetic stripe reader is not output to the console but notified only to the application that needs the data. If you do not use the keyboard hook driver, the magnetic stripe reader and keylock data is processed as usual keyboard data. Therefore, even if you do not want to display the data of the magnetic stripe reader and keylock, run "kbh_load" to start the keyboard hook driver and the hooking process.

4.2 Hardware settings

4.2.1 Setting devices

The devices supported by EPSON JavaPOS ADK can be set in various way by using the dip switches and so on. However there are some restrictions on operating EPSON JavaPOS ADK. See "Appendix-A Device Settings" for details of the settings.

4.2.1.1 Setting the POSPrinter

The following restrictions apply.

- Use large capacity for the receive buffer.
- Use DTR/DSR for handshake (flow control).
- Use 8 bits for the bit length.
- Setting offline or receive buffer full as a Busy condition is recommended[Note1]

[Note1] If the device that cannot send data with condition Busy is used with a Ethernet interface board, set receive buffer full as the busy condition.

4.2.1.2 Setting line display

The following restrictions apply.

- Use DTR/DSR for handshake (flow control).
- Use 8 bits for bit length.
- Set 1-2 of jumper switch1 (JP1) short when you use the DM-D110 and DM-D210 with the option stand.

4.2.1.3 Setting MSR

Check that each item is set correctly using the utility included with each device. The devices do not operate except for the following setting. Refer to the manual included with each device for operation of the utility.

(1) EPSON DM-MR123

The DM-MR123 can be installed on the IM-310 as an option.

Track	Header information	Footer information
1	Yes ("%")	Yes ("?")
2	Yes (";")	Yes ("?")
3	Yes ("+")	Yes ("?")

Readable Track : Track1, Track2, Track3
MSR code type : 101 keyboard

(2) EPSON DM-MS123

DM-MS123 can be installed on the IM-600 as an option.

Track	Header information	Footer information
1	Yes ("%")	Yes ("?")
2	Yes (";")	Yes ("?")
3	Yes ("+")	Yes ("?")

Readable Track : Track1, Track2, Track3
MSR code type : 101 keyboard

(3) EPSON DM-MR112

DM-MR112 can be installed on the IM-310 as an option.

Track	Header information	Footer information
1	Yes ("%")	Yes ("?")
2	Yes (";")	Yes ("?")
3(JIS-II data)	Yes (" ")	Yes (" ¥R")

Readable Track : Track1, Track2, JIS-II
MSR code type : 101 keyboard

(4) EPSON DM-MS112

DM-MS112 can be installed on the IM-600 as an option.

Track	Header information	Footer information
1	Yes ("%")	Yes ("?")
2	Yes (";")	Yes ("?")
3(JIS-II data)	Yes (" ")	Yes (" ¥R")

Readable Track : Track1, Track2, JIS-II
MSR code type : 101 keyboard

4.2.1.4 Settings keylock

(1) IM-310

Check that each item is set correctly using the utility included with the IM-310. The IM-310 does not operate with settings other than the following. Refer to the manual included with IM-300 for operation of the utility.

Key position	Scanning code
0	64h
1	65h
2	66h
3	67h
4	68h
5	69h
6	6Ah

4.2.2 Setting the EPSON IM series

4.2.2.1 Setting the IM-310

When the TM-T88IIR or DM-D display is connected to the IM-310, make sure that the jumper switches are set as follows to specify the flow control of the devices.

- Set 2-3 of JP32 on the main board short
- Set 2-3 of JP1 besides the customer display short

Refer to the IM-310 product guide for locations of the jumper switches.

4.2.2.2 Setting the IM-600

When you use two drawers on the IM-600, the drawer boards and COM3 should be set to the native mode. Make sure that the BIOS of the IM-600 and dip switches of the drawer board are set as follows:

- Set "Serial Port 3 Mode" in the BIOS of the IM-600 DRW/DM-D
- Set the dip switches on the drawerboard as follows:1-1 to OFF and 1-2 to ON

The drawer to connect the IM-600 should use pin number 2 to notify the open/close signal. Also the open signal level of the drawer should be the same as the IM-600’s port. However, when you set "connect pin number" of the drawer using the EJEE utility (see "4.3.3.9 Editing the device details of cash drawer"), the pin number changes by the connecter that the drawer is connected to. Set the pin number as follows:

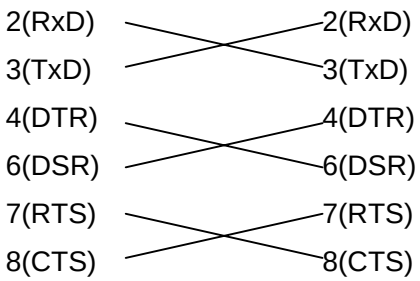
- Set "connect pin number" 2 when the drawer is connected to connecter A
- Set "connect pin number" 5 when the drawer is connected to connecter B

4.2.3 Available cable

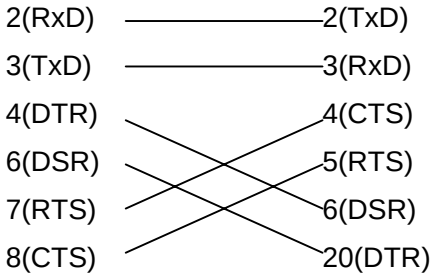
4.2.3.1 Serial cable

Use the serial cable with the following connection forms.

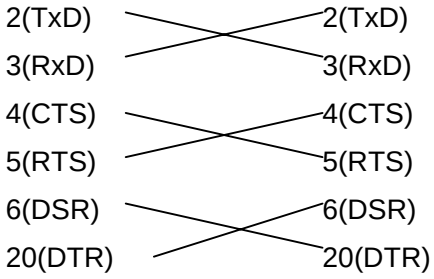
a) For D-SUB9 to D-SUB9



b) For D-SUB9 to D-SUB25



c) For D-SUB25 to D-SUB25



4.2.3.2 Ethernet cable

Ask the administrator of your network for the details. Select 10BASE-T or higher (e.g. 100BASE-TX), category 5 cable for normal use. Use a straight or crossed cable that fits your network environment. Normally, straight cable is used on a network with hub. The crossed cable is used on a network where the computer and devices are connected directly.

4.2.4 Setting the UB-S09

The UB-S09 board is used only for connecting the line display to the POS printer without the connector for customer display.

Set the UB-S09 as follows:

- Set 1-2 short on the JP1
- Set 1-2 short on the JP2

4.3 Setting EPSON JavaPOS ADK (EJEE utility)

When using EPSON JavaPOS ADK, you need the jpos.xml file, which contains the operation environment of EPSON JavaPOS ADK. To create or edit the jpos.xml file. use the EJEE (EPSON JavaPOS Entry Editor) utility. In this chapter, the jpos.xml file is indicated as a configuration file.

The EJEE utility is used to create or edit configuration files of EPSON devices. It is based on "JposEntryEditor2.0.1" included in JCL2.0.1. The configuration files of devices other than EPSON devices can be created and edited, and the operation follows the specification of "JposEntryEditor2.0.1".

This section describes how to set EPSON devices using the EJEE utility.

4.3.1 Starting the EJEE utility

The EJEE utility is installed in the directory specified in "3.4.3 Installing EPSON JavaPOS ADK and EJEE". Java2 SDK is required when using the EJEE utility.

To start EJEE utility, move to the directory that EJEE utility is installed and run the following command.

./ejee

4.3.2 Setting a configuration file

4.3.2.1 Reading a configuration file

The configuration file must exist in the EJEE utility. By default, this file is not loaded when the EJEE utility starts, but you can change the EJEE utility setting so that this file is read automatically when the EJEE utility starts. See "4.3.3.3 Operation option of the EJEE" for details. However, if you are using Redhat 7.1, do not use automatic reading of the configuration file or the EJEE utility cannot start.

4.3.2.2 Saving a configuration file

The configuration file created and edited with the EJEE utility is saved in the directory that the EJEE utility is stored.

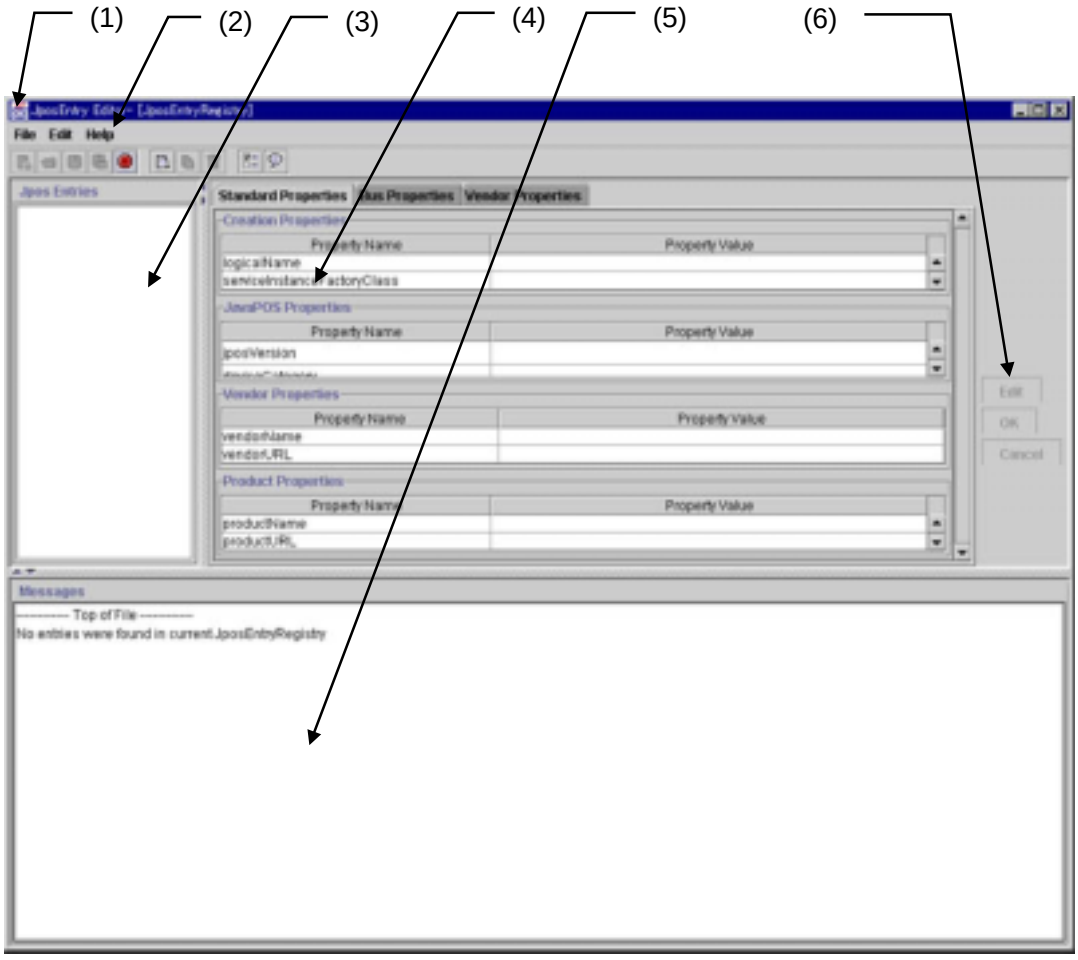
When you use EPSON JavaPOS ADK, copy the saved configuration file to the directory specified in "Specifying a path to jpos.xml to read at when JavaPOS starts" of "3.4.3.3 Installation procedure". If you have not specified the directory, copy the file to the directory where the application starts.

4.3.3 Using the EJEE utility

In this section, the setting item for the device is indicated in bold letter. Make sure the item is same as the setting of the device itself.

4.3.3.1 Main screen

When the EJEE utility starts, the following window (main screen) appears.



(1) Control menu buttons

Minimize the EJEE utility, close it, and so on.

(2) Menu bar

Indicates lists of commands, if you select [File], [Edit] and so on.

(3) Entry display tree

Indicates the tree that shows the current registered entries (devices).

(4) Property display tab

Indicates the property information of entries selected in the entry display tree.

(5) Message area

Indicates EJEE utility messages. The messages follow "JposEntryEditor2.0.1".

(6) Property edit buttons

Use them to edit the property value.

4.3.3.2 Reading and saving the configuration file

To read the configuration file into the EJEE utility, select [Read JposRegistry] from the [File] menu.

To save the configuration file, select [Save JposRegistry] from the [File] menu.

4.3.3.3 Operation option of the EJEE utility

When you select [Option] from the [File] menu, the following dialog box appears. You can use this dialog box to change the operation of the EJEE utility.

The following items in the dialog box can be changed.

(1) [Tree View] tab

- Changes the display form of the entry display tree.
- Sets the automatic expansion of the entry display tree.

(2) [Look and Feel] tab

- Changes the user interface of the whole EJEE utility.

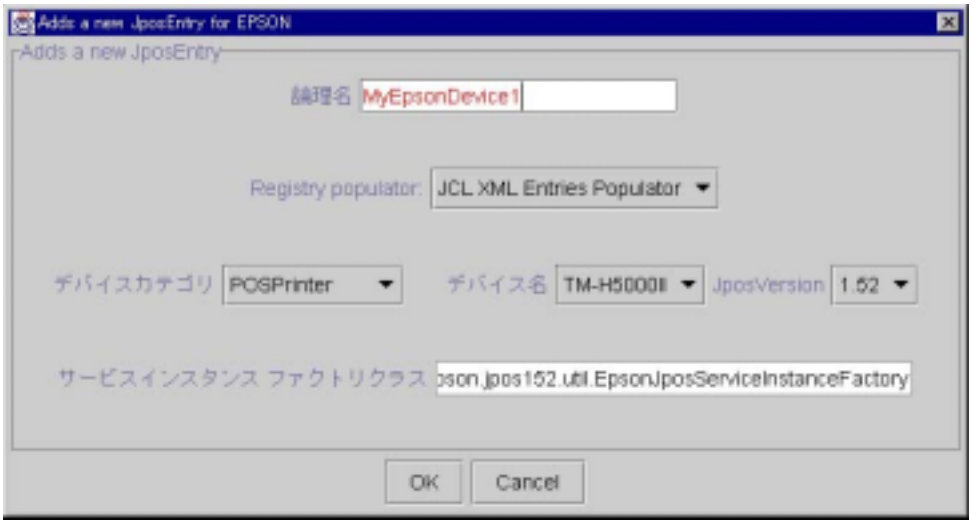
(3) [Other] tab

- Sets the value to hex representation.
- Sets automatic reading of configuration.
- Switches of the operation when an entry is copied.

4.3.3.4 Adding new devices

Use the following procedure to add a new device to the entry.

(1) Select ADD EPSON from the [File] menu.
The following dialog box appears.



(2) In the [Logical Name] text box, input a logical name to register to the entry. Be sure not to input a name that has already been registered in the entry. The blue characters in the [Logical Name] text box indicate that the same logical name has already been entered.
(3) Select a device category to register from the [Device Category] combo box. When the device category is changed, the [Device Name] combo box also changes to a list of devices that can be registered.
(4) Select a device to register from the [Device Name] combo box.
(5) Select a version number of JavaPOS to register from the [JposVersion] combo box.
(6) Click [OK] to register the setting to the entry, or [Cancel] to abort the operation.

When you add a new entry, the property of the device is registered with the default value.

4.3.3.5 Editing an entry

Use the following procedure to change the property of a registered entry.

(1) From the entry display tree, select the entry you want to edit.
(2) Select the [Vendor Properties] tab from the property display tab.
The property of the EPSON devices can be edited only from the [Vendor Properties] tab but not the other tabs.
(3) Click [Edit].
The following dialog box appears.

(4) Common property of all devices can be edited in the [Common] tab.
The logical name, the device to use, and the version of JavaPOS to use are changed in the [Important property] panel. This tab can be used in the same way as the dialog box showed when a new device is added. See "4.3.3.4 Adding new devices" for details.

If you change the device category or the device name, the all property value is canceled and it returns to the default value.

Although a property value is inherited when the version of JavaPOS is changed, when the version is dropped, the communication interface related property may be initialized to the serial interface.

(5) The specific property of the device currently selected can be edited in the [Device Details] tab. This tab does not appear if the selected device does not have its specific property. Items to edit in the [Device Details] tab are different depending on the device.

(6) The property related to the communication interface can be edited in the [Port] tab except for the keyboard interface. Items to edit in the [Port] tab are different depending on the device.

This tab does not appear for keyboard interface devices.

(7) Click [OK] to save the edited setting, or [Cancel] to cancel them.

4.3.3.6 Editing the property related to communication interface

The device property related to the communication interface can be edited in the [Port] tab of the edit dialog box of the entry.

The following procedure shows how to edit in the [Port] tab.

(1) **In the [Physical Port Name] text box, input the name of the port that the device is connected to.**
If the connection status is "Serial", input "/dev/ttyS0" or "/dev/ttyS1".
If the connection status is "Ethernet", input the device IP address.
If the connection is "USB", you can input the port name in the following three ways.

- **Input the serial number of the interface board of the device.**
The serial number is written in the "device" file stored in the /proc/bus/usb directory.
- **Input "usb"**
With this method, JavaPOS searches for the USB interface device automatically. However, if more than two EPSON POS devices with USB interface are connected, JavaPOS cannot find them. In this case, do not input "usb".
- **Input the printer model (default)**
With this method, JavaPOS searches for the USB interface device automatically. However, if two or more of the same device with USB interface are connected, JavaPOS cannot find them. In this case, input the serial number of the interface board.

(2) **Specify the type of communication interface of the device in the [Communication Type] combo box.**
If you change the communication interface, the all property value related to the communication interface goes back to the default value.

(3) In the [Time to retry[ms]] textbox, input the time to retry sending to the communication interface. The time can be set to a value in the range of 1 to 99999 milliseconds.

(4) In the [Data Transmission Timeout [ms]] textbox, input the time that JavaPOS determines an error when the transmission to the communication interface fails. The time can be set to a value in the range of 1 to 99999 milliseconds. Set a value larger than the value of the Time to retry[ms] textbox.

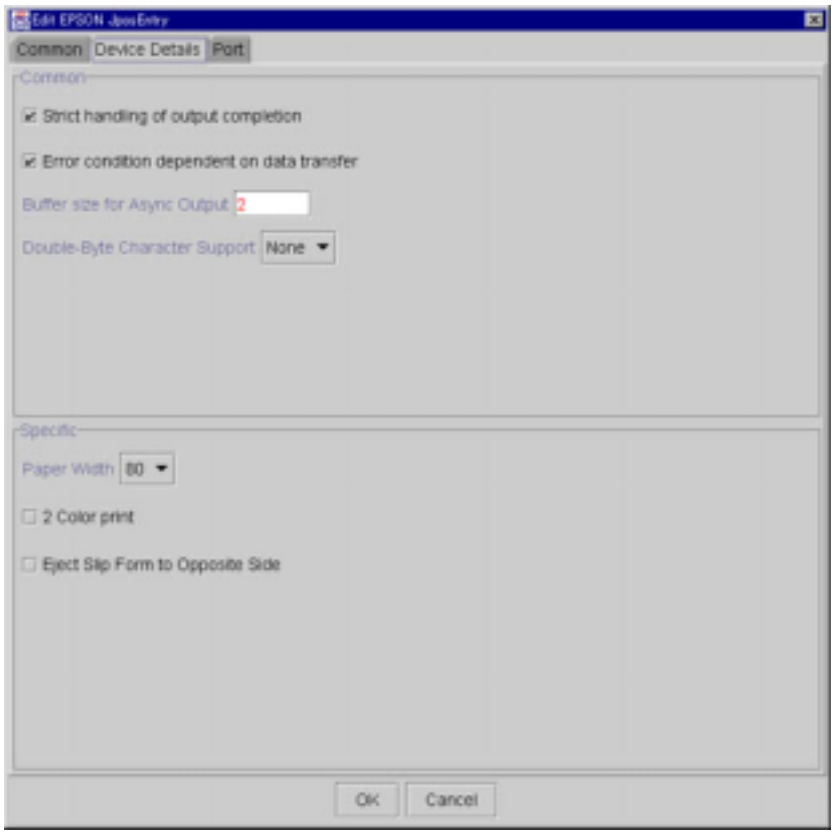
(5) Input the time to timeout receiving from the communication interface in the [Data Reception Timeout[ms]] textbox. The time can be set to a value in the range of 1 to 99999 milliseconds.

The communication condition of the serial devices can be edited in the Serial Specific panel. This panel can be edited only when "Serial" in the Communication Type combo box is selected. Make sure that the setting of the [Serial Specific] panel is same as the communication condition of the device.

(1) **Specify the communication speed in the [Baud Rate] combo box.**
(2) **Specify the type of parity check in the [Parity] combo box.**
(3) **Specify the type of flow control in the [Flow Control] combo box**
(4) **Specify the data bits length in the [Data bits] combo box.**
(5) **Specify the stop bits length in the [Stop bits] combo box.**
(6) Input the number of data that can be transmitted in a single transmission to the communication interface in the [Length of single output[bytes]]text box. The number of data can be set to a value in the range of 1 to 99999 bytes. If you set a larger value, the transmission speed becomes higher. However, if you set it too large, some of the data is lost during transmission. The maximum value is 40% of the capacity of the device buffer that is full and busy (depending on the conditions).

4.3.3.7 Editing the device details of the POS printer

The following figure shows [Device Details] of the POSPrinter. Items indicated in the dialog box are different, depending on the device. (All items are displayed in the following dialog box.)



(1) Select the [Strict handling of output completion] check box to make the completion time of the print method same as the time of the print completion. If you clear the check box, the completion time of the print method becomes the same as the completion time of data transmission. If the device selected as the serial/USB interface supports power recovery function, the device always operates as if the function was ON, regardless of the state of this check box.

(2) Select the [Error condition dependent on data transfer] check box to determine when the error has occurred after performing the print method: the time before data transmission or the time during data transmission or data printing.
If you clear the check box, the error occurs in both cases.
If the [Strict handling of output completion] check box is cleared or the device selected as the serial/USB interface supports the power recovery function, the device always operates as if the function was OFF, regardless of the state of this check box.

(3) In the [Buffer size for Async Output] text box, input the amount of data per transfer sent to the printer for asynchronous output. This setting is effective only if the device selected as the serial/USB interface supports the power recovery function. The value can be from 1 to 100. If you set a larger value, the performance of asynchronous input becomes better. However, data after a transaction causing an error may be printed when the error occurs.

(4) **In the [DoubleByte Character Support] combo box, specify the type of double-byte character of the device installed.**

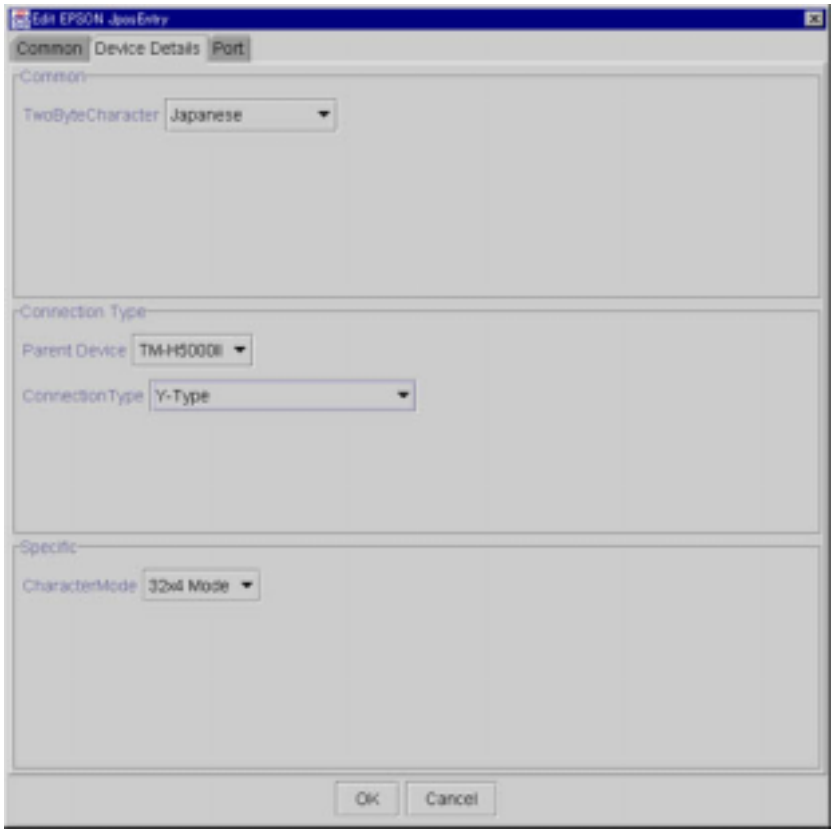
(5) Specify the paper width in the [Paper Width] combo box. For the TM-L90, the combo box changes to text box for inputting the paper width. You can specify a value in the range of 38 to 80 mm.

(6) Select the [2 Color print] check box to enable 2-color printing.
For the TM-U210A and TM-U230, select this check box if the ink cartridge that can print two colors.

(7) Select the [Eject Slip Form to Opposite Side] check box to eject slip form to the opposite side.

4.3.3.8 Editing the device details of the line display

The following figure shows [Device Details] of the line display. Items indicated in the dialog box are different, depending on the device. (All items are displayed in the following dialog box.)



(1) **In the [TwoByte character] combo box, specify the type of double-byte character of the device installed.**

(2) **In the [Parent Device] combo box, specify the POS printer that the device is connected to. Specify the "TM-T88IIR" when you connect the customer display to the device.**

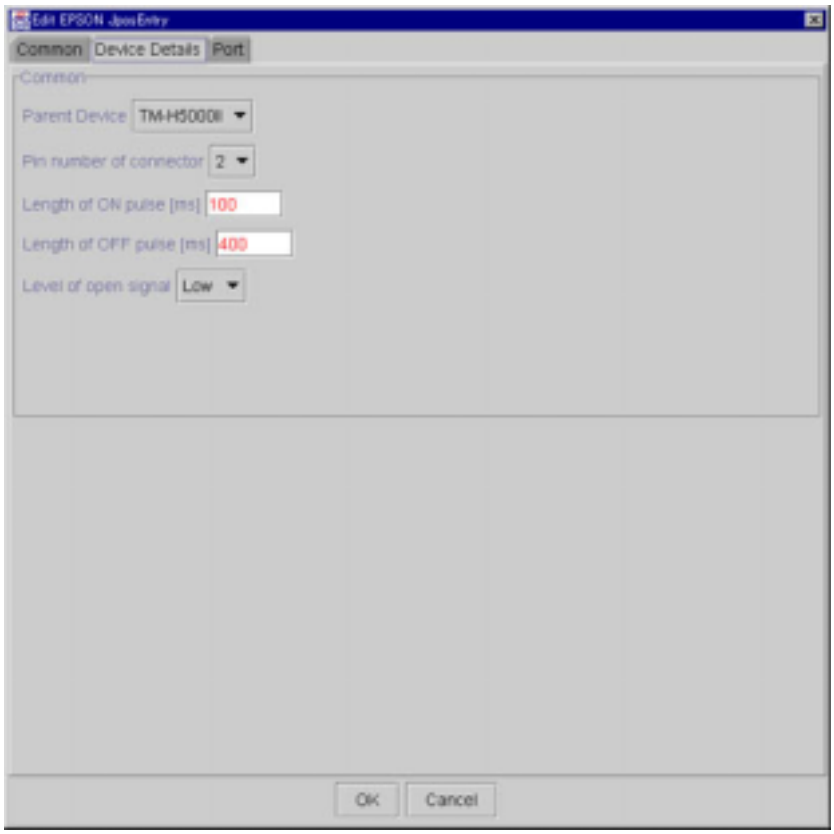
(3) **Specify the connection type of the devices in the [Connection Type] combo box.**
"Stand alone" indicates a connection type where the device is connected to the host computer without any devices connected to it.
"Y-connection" indicates a connection type where the POS printer is connected to the host computer and a device or the interface board of customer display is connected to the POS printer.
"Pass-through" indicates a connection type where the device is connected to the host directly and the POS printer is connected to the device. Depending on the device, you can set "Y- connection" for this connection type; but in this case, select “Pass-through” in the EJEE setting.
If the device is connected to the customer display of the IM-600, select "IM-600".

Note: In the previous version, you can select "Y-connection using the optional stand" in the [Connection Type] combo box; however, this option is deleted in this version. When you import the setting file created in the previous version, the item set as "Y-connection using the optional stand" is converted to “Pass-through”.

(4) Specify a type of display screen of the line display in the [CharacterMode] combo box.

4.3.3.9 Editing the device details of cash drawer

The following figure shows [Device Details] of the cash drawer.



(1) **In the [Parent Device] combo box, specify the POS printer that the device is connected to.**
Specify "TM-T88IIR" if the device is connected to the customer display of the IM-310.
Specify "IM-600" if the device is connected to the customer display of the IM-600.

(2) **In the [Pin number of connector] combo box, specify a pin number that notifies the open/close signal of the drawer in.**

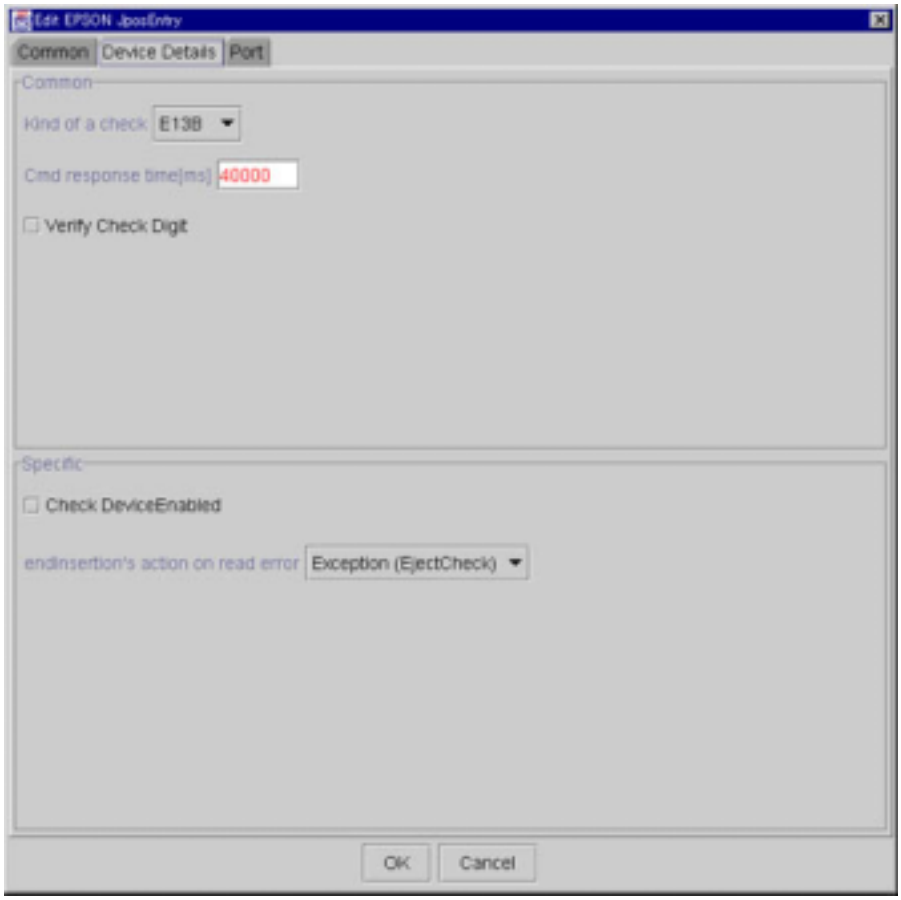
(3) In the [Length of ON pulse[ms]] text box, input the length of ON pulse when the drawer is opened. The length of ON pulse can be set to a value in the range of 50 to 999 milliseconds.

(4) Input a length of off pulse when the drawer is closed in the [Length of OFF pulse[ms]] text box. The length of off pulse is set in milliseconds from 50 to 999 milliseconds.

(5) **Specify a signal level of pulse when the drawer is open in the [Level of open signal] combo box.**

4.3.3.10 Editing the device details of MICR

The following figure shows [Device Details] of the MICR. Items indicated in the dialog box are different, depending on the device. (All items are displayed in the following dialog box.)



(1) In the [Kind of a check] combo box, select a type of check to use.

(2) In the [Cmd response time[ms]] combo box, specify the time for timeout from insertion of check to receiving check data. Timeout can be set to a value in the range of 1 to 999999 milliseconds.

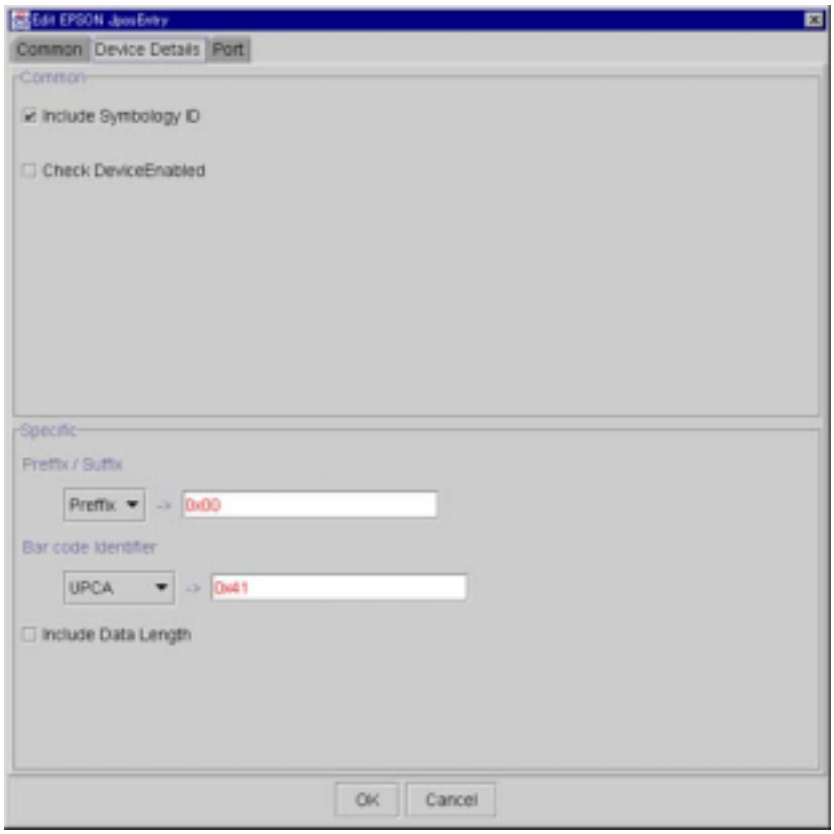
(3) If you select the [Verify Check Digit] check box, the check digit of transmit number field of the E13B check is checked.
The check box becomes disabled if the CMC7 is checked.

(4) If you select the [Check DeviceEnabled] check box, the DeviceEnabled property is verified when DataEvent is issued. If the property is false, DataEvent is not issued until the property becomes true.

(5) In the [endInsertion's action on read error] combo box, select the action on the read error of check when the endinsertion method is performed.
If you select "Exception", the endInsertion method notifies an exception when the read error of check occurs. The check is not ejected even when the error occurs.
If you select "Exception(EjectCheck)", the endInsertion method notifies an exception when the read error of check occurs. The check is ejected when the error occurs.
If you select "ErrorEvent", ErrorEvent notifies the error and the endInsertion method ends normally. The check is not ejected even when the error occurs.

4.3.3.11 Editing the device details of the scanner

The following figure shows [Device Details] of the scanner. Items indicated in the dialog box are different, depending on the device. (All items are displayed in the following dialog box.)



(1) **In the [Include SymbologyID] check box, specify whether the symbology is included to the bar code data sent from the device. If the symbology ID is included, select check box. In this case, set the DecodeData property to true, to determine the type of bar code.**

(2) If you select the [Check DeviceEnabled] check box, the DeviceEnabled property is verified when DataEvent issued is . When this property is false, DataEvent is not issued until the property becomes true.

(3) **Select an item (Prefix or Suffix) to edit in the [Prefix/Suffix] combo box Prefix/Suffix. Input the code included if the device sends the bar code data in the text box beside the combo box. Input the code in ASCII (hexadecimal) and be sure to start with "0x" to indicate a hexadecimal**

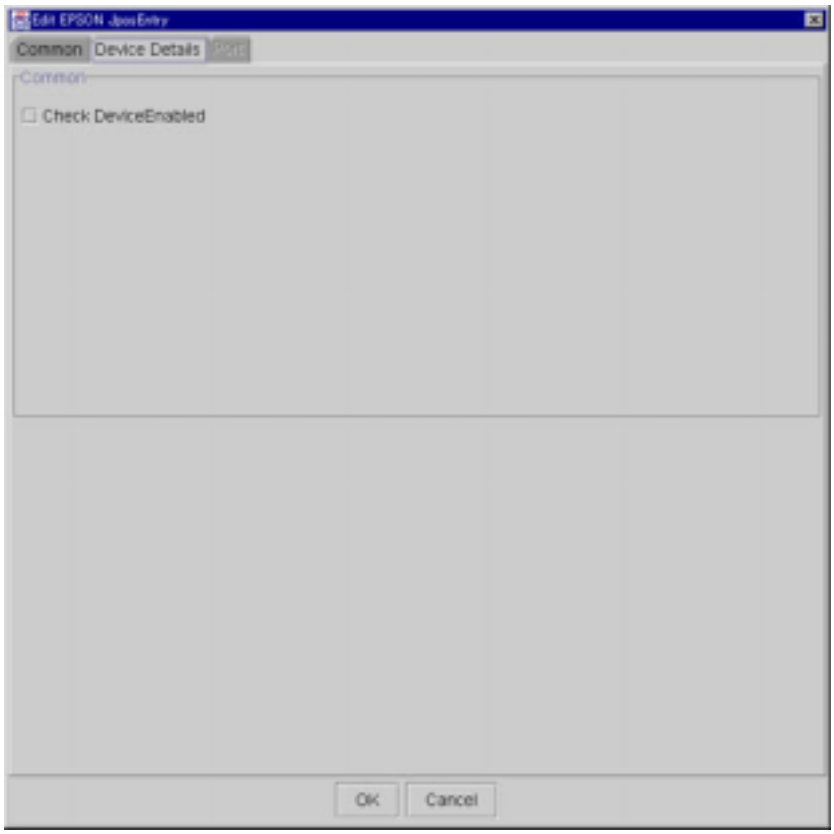
number. When inputting two or more bytes, separate them with comma ",".
Input "0x00" if the code of the selected item is not included. Be sure to add the Suffix including the setting of the device itself. JavaPOS does not operate correctly if the Suffix is not set.

(4) **Select an item (type of bar code) to edit in [Bar code Identifier]. Input the bar code identifier that is included when the device sends bar code in the text box beside the combo box. Input the code in ASCII (hexadecimal) and be sure to start with "0x" to indicate a hexadecimal number. When inputting two or more bytes, separate them with comma ",".**
Input "0x00" if the code of the selected item is not included.

(5) **The [Include Data Length] check box pecifies whether the bar code data sent from the device includes the data length.**
Select the check box if the data length is included.

4.3.3.12 Editing the device details of MSR

The following figure shows [Device Details] of the MSR. Items indicated in the dialog box are different, depending on the device. (All items are displayed in the following dialog box.)



(1) If you select the [Check DeviceEnabled] check box, DeviceEnabled is verified when DataEvent is issued. When this property is false, DataEvent is not issued until the property becomes true.

4.4 Setting JCL

To operate JavaPOS, the JCL(JavaPOS Configuration/Loader) Version2.0.1 package is required. In EPSON JavaPOS ADK, the package is included in the "epsonJposControl15.jar" file that is expanded during the installation.

JCL provides classes to access the configuration file that is required when JavaPOS operates, as well as loader classes that are required when DeviceControl loads DeviceService.

JCL needs the configuration file to operate. The location and name of the configuration file must be written in the jpos.properties file of the JCL package.

When you install EPSON JavaPOS ADK, update "jpos.properties" by inputting "jpos.xml" as the name of the configuration file and a location of the configuration file, following "Step2: Specifying a path to jpos.xml to read at when JavaPOS starts" of "3.4.3.3 Installation procedure".

If EPSON JavaPOS ADK is not used in the environment (described above) that is specified in its installation, the "jpos.properties" file in the "epsonJposControl15.jar" file should be changed. The following procedure shows how to change the "jpos.properties" file. Java2 SDK is required for this procedure.

[Procedure]
Step1: Moving to the directory
Move to the directory that the "epsonJposControl15.jar" is stored.

Step2: Expanding "jpos.properties"

Expand the "jpos.properties" file in the "epsonJposControl15.jar" file only.

jar x jpos/res/jpos.properties < epsonJposControl15.jar

The expanded file is stored in the jpos/res/ directory under the current directory.

Step3: Editing "jpos.properties"

Edit the "jpos.properties" file using the editor tool. The "jpos.config.populatorFile" key on the 10th line should be changed. Input the name of the configuration file including its directory name. The directory can be either a full or relative path name. For the relative path, the directory where the application (that uses JavaPOS) starts is the current directory.

Step4: Compressing "jpos.properties"

Compress "epsonJposControl15.jar" after editing the "jpos.properties" file. Move to the directory that the "epsonJposControl15.jar" file is stored and then execute the following command.

jar uf epsonJposControl15.jar jpos/res/jpos.properties

Step5: Deleting "jpos.properties"

Delete the file expanded in Step 2.

rm –r –f jpos

Chapter 5 Example of execution

This chapter explains how to run JavaPOS, using a sample program included with EPSON JavaPOS ADK. The directory that EPSON JavaPOS ADK is stored is indicated as "JPOS_HOME" in this chapter.

5.1 Execution environment

5.1.1 Setting the Java environment

The following packages are needed to execute the sample program.

- Java2 SDK
- RxTx (if the device uses serial I/F)
- EPSON USB driver (if the device uses with USB I/F)
- EPSON JavaPOS ADK

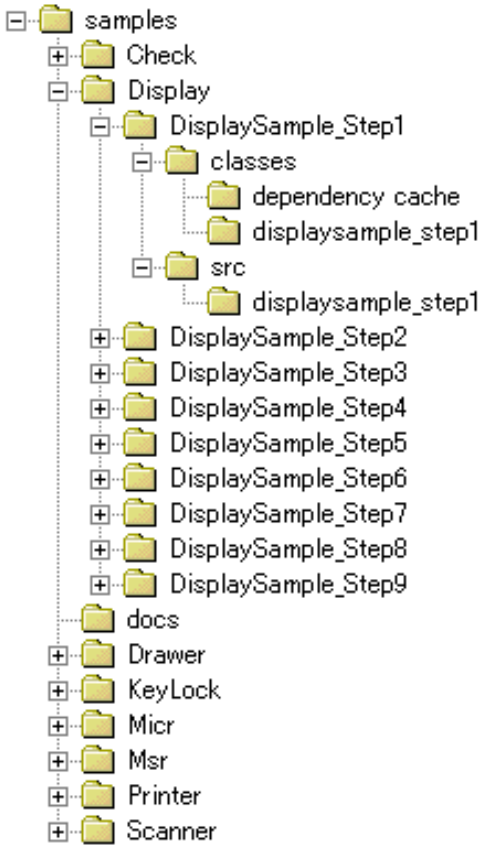
Set Java2 SDK and RxTx according to "Chapter 3 Installation", before executing the sample program.

5.1.2 Expanding the sample program

The sample program is compressed as the "EPSON_Jpos_Samples.tar.gz " file in the <*JPOS_HOME*>/javapos directory. Expand this file in the directory that the file is located.

tar –xvzf EPSON_Jpos_Samples.tar.gz

The sample program is expanded in the following directory.



5.1.3 Setting JavaPOS

If the default setting is used in "Step2: Specifying a path to jpos.xml to read at when JavaPOS starts" of "3.4.3.3 Installation procedure", copy the "jpos.xml" file to the following directory.

<*JPOS_HOME*>/javapos/samples

If you have set the path, copy the "jpos.xml" file to the specified directory.

Set the logical name in the EJEE utility as follows:

Device	Logical name
POSPrinter	"Printer"
Line Display	"Display"
Cash Drawer	"Drawer"
MICR	"MICR"
CheckScanner	“CheckScanner”
Scanner	"Scanner"
MSR	"MSR"
Keylock	"Keylock"

5.2 Execution of the sample program

5.2.1 Executing the sample program

Execute the sample program using "run_sample.sh" in the <*JPOS_HOME*>/javapos/samples directory.
Run the following command.

sh run_sample.sh *Device StepNo*

Input the device that executes the sample program in the Device. Input the step number of the sample program in *StepNo, as follows:*

Device	***Device***	***StepNo***
POSPrinter	Printer	1 to 11
Line Display	Display	1 to 9
CashDrawer	Drawer	1 to 3
MICR	MICR	1 to 5
CheckScanner	CheckScanner	1 to 5
Scanner	Scanner	1 to 3
MSR	MSR	1 to 3
Keylock	Keylock	1 to 3

5.2.2 Compiling the sample program

If you want to execute the sample program after editing it, change the code of basic step then execute the "make_package" file in the src directory of the step.

./make_package

Executing the "make_package" file compiles the source file. The jar file is then created and copied to the <*JPOS_HOME*>/javapos/samples directory.
If you want to execute the program after the compilation, follow the procedure explained in "5.2.1 Executing the sample program".

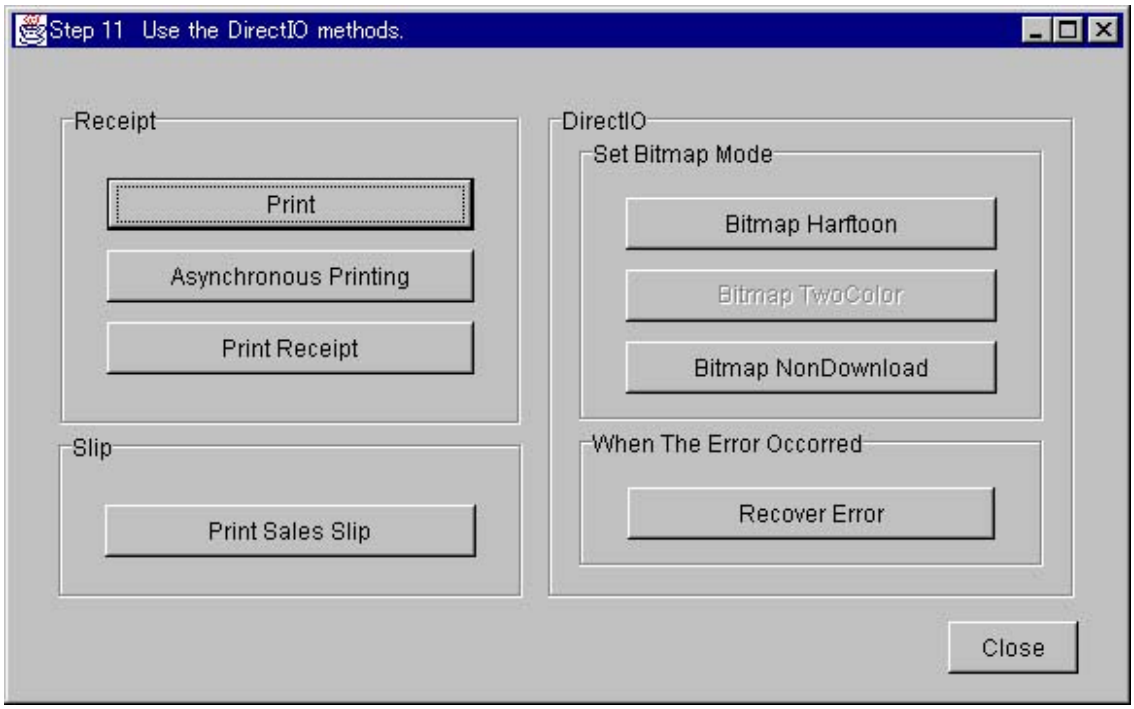
5.3 Outline of the sample program for each device

This section gives an outline of the sample program for each device. Refer to the "EPSON JavaPOS ADK operation manual" for details of the sample program.

5.3.1 POSPrinter

The sample program of the POSPrinter provides 11 steps, starting from simple printing to using the directIO method.

This section explains the sample program, using step11 as an example.



The explanation of the process of each button is as follows:

[Print] button
Prints the sample on the receipt.

[Asynchronous printing] button
Make the printing process asynchronous so the process returns even when printing is not completed.
When printing is completed, a message dialog appears to indicate printing is

completed.

[Print Receipt] button
Prints a sample of the receipt, using rotation print.

[Print Sales Slip] button
Prints a sample on the slip.

[Bitmap Halftone] button
Prints an image using half tone.

[Bitmap TwoColor] button
Prints an image using two colors.

[Bitmap NonDownload] button
Registers an image in the setBitmap method using the download function.

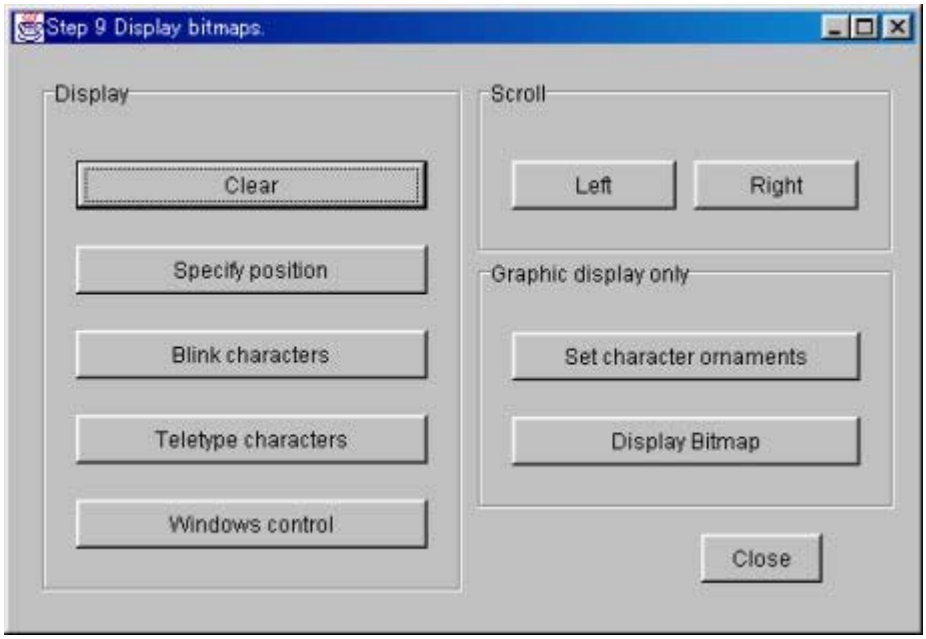
[Recover Error] button
Recovers from a recoverable error.

[Close] button
Ends the sample program.

5.3.2 Line Display

The sample program of the Line Display provides 9 steps, starting from simple display to using the graphic function.

This section explains the sample program, using step 9 as an example.



The explanation of the process of each button is as follows:

[Clear] button
Clears the display.

[Specify Position] button
Specify the starting position of the display as well as the display characters.

[Blink Characters] button
Flashes characters.

[Teletype Characters] button
Displays teletype characters.

[Window Control] button
Makes a window and displays characters with marquee.

[Left] button
Scrolls the displayed character to the left.

[Right] button
Scrolls the displayed character to the right.

[Set Character Ornaments] button
Displays ornamentation characters using escape commands extended by EPSON specification(Graphic Display only).

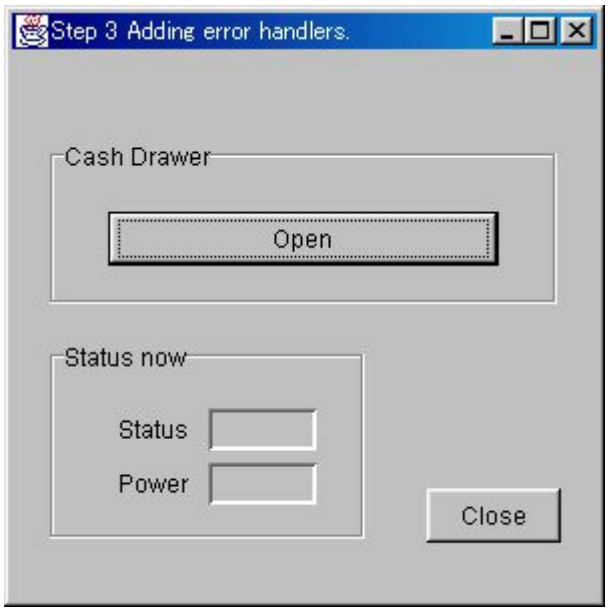
[Display Bitmap] button
Displays an image(Graphic Display only).

[Close] button
Ends the sample program.

5.3.3 CashDrawer

The sample program of cash drawer provides 3 steps, starting from opening the drawer opening to exception handling.

This section explains the sample program, using step 3 as an example.



The explanation of the process of each button is as follows.

[Open] button
Opens a drawer.

[Close] button
Ends the sample program.

The status of a drawer appears in the following text area.

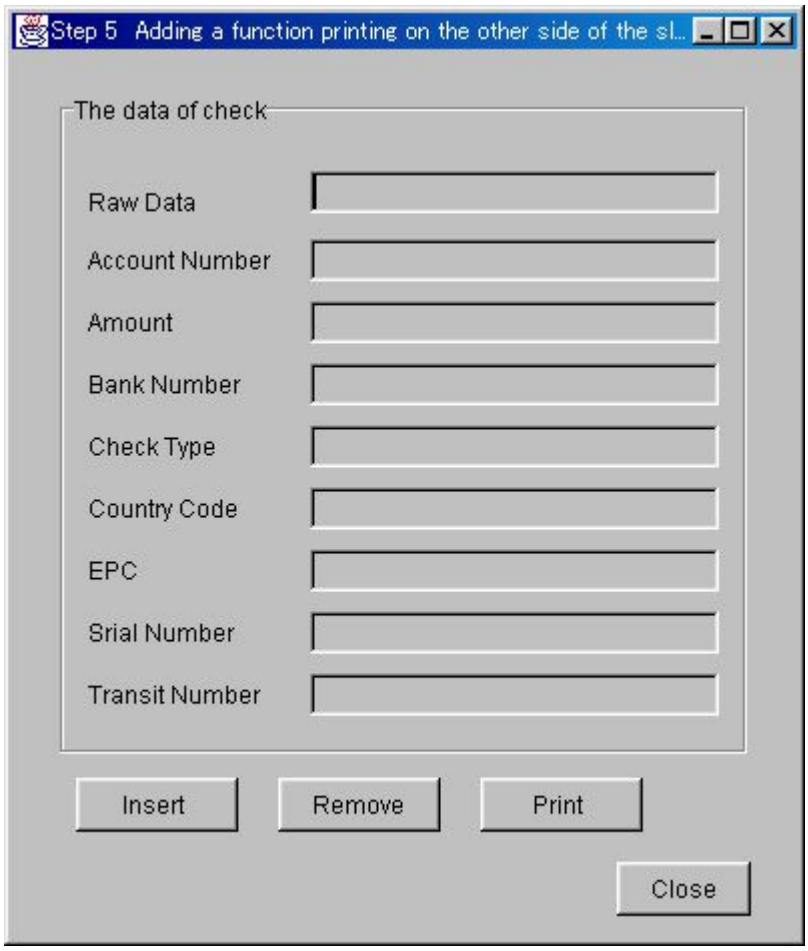
[Status] text area
Displays the current status (Open/Close) of the drawer.

[Power] text area
Displays the current power status of the drawer.

5.3.4 MICR

The sample program of the MICR provides 5 steps, starting from reading a check reading to endorsing printing.

This section explains the sample program, using step 5 as an example.



The explanation of the process of each button is as follows:

[Insert] button
Inserts and reads a check.

[Remove] button
Ejects a check.

[Print] button

Prints a message that indicates proof on the back of a check.

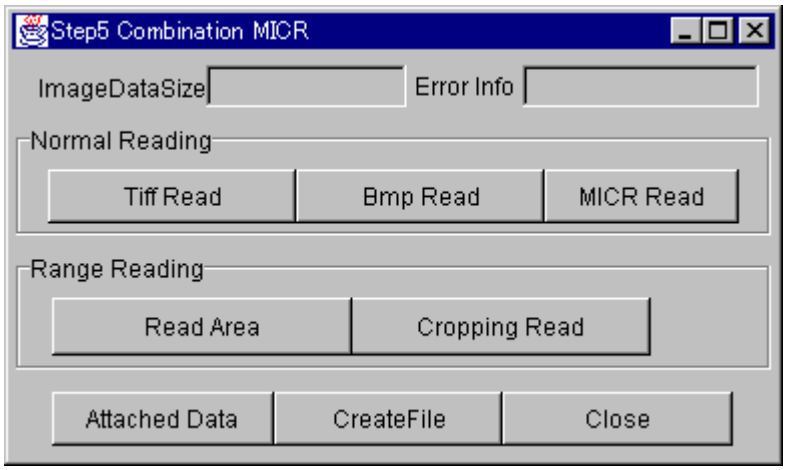
[Close] button
Ends the sample program.

The information of the check appears in each of the text areas.

5.3.5 CheckScanner

The sample program of the POSPrinter provides five steps: from check scanning to combination with MICR.

This section describes the sample program, using step 5 as an example.



The explanation of the process of each button is as follows:

[Tiff Read] button
Inserts a check and then scans in the TIFF file format.

[Bmp Read] button
Inserts a check and then scans in the BMP file format.

[MICR Read] button
Inserts a check and reads using MICR.

[Read Area] button

Inserts a check, sets the read area and then scans in the BMP file format.

[Cropping Read] button
Inserts a check, sets the Cropping area and then scans in the BMP file format.

[Attached Data] button
Inserts a check and then scans in the TIFF file format using the attachedData method.

[CreateFile] button
Saves image data to the file.

[Close] button
Closes the sample program.

The following text areas show the scan results.

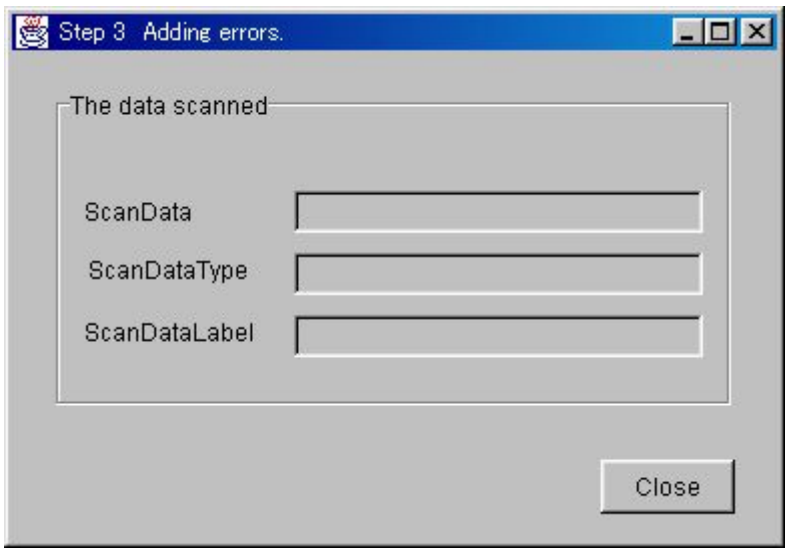
[ImageDataSize] text area
Displays the scanned image data size (in bytes).

[Error Info] text area
Displays the reading failure of CheckScanner or MICR.

5.3.6 Scanner

The sample program of the Scanner provides 3 steps, starting from barcode reading to exception handling.

This section explains the sample program, using step 3 as an example.



The explanation of the process of each button is as follows:

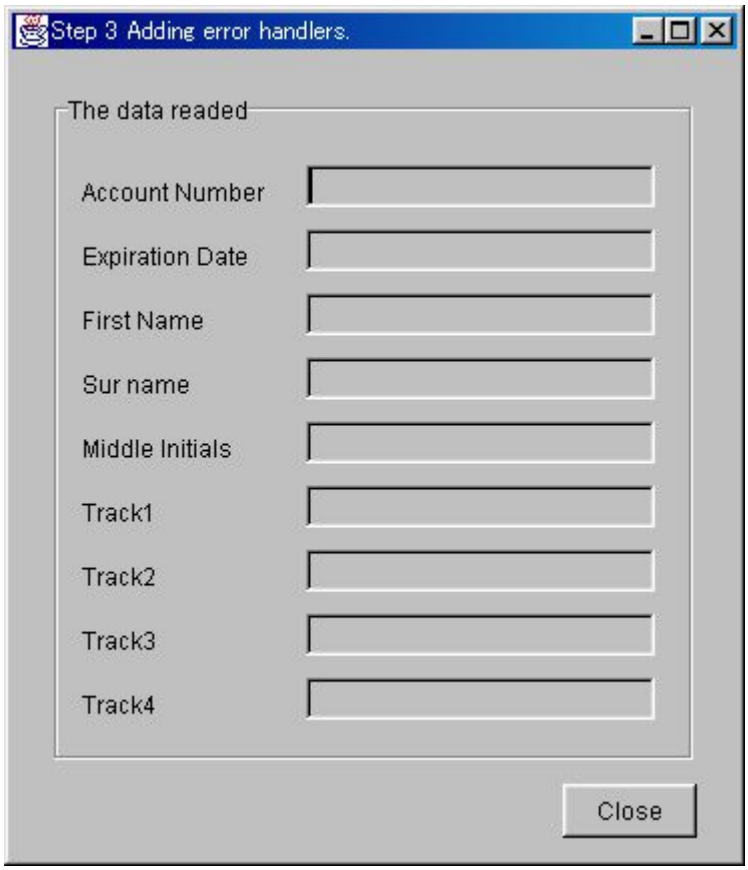
[Close] button
Ends the sample program.

The information of the barcode appears in each of the text areas.

5.3.7 MSR

The sample program of the MSR provides 3 steps, starting from card reading to exception handling.

This section explains the sample program, using step 3 as an example.



The explanation of the process of each button is as follows:

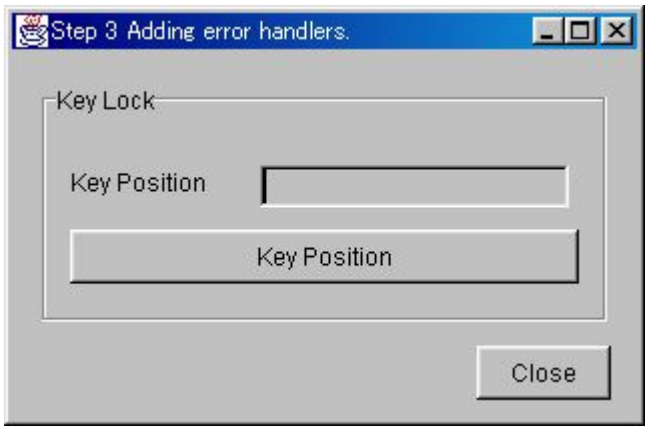
[Close] button
Ends the sample program.

The information of the card appears in each of the text areas.

5.3.8 Keylock

The sample program of Keylock provides 3 steps, starting from acquisition of key position to exception handling.

This section explains the sample program, using step 3 as an example.



The explanation of the process of each button is as follows:

[Key Position] button
Displays a message dialog box and waits for change of the key position.

[Close] button
Ends the sample program.

The current key position appears in each of the text areas.

Chapter 6 Trace

With EPSON JavaPOS ADK, you can record the interaction between the device control and the device service, as text in a log file. This is called a trace function.

This chapter describes how to use the trace function, as well as format of the log file.

6.1 Special notice on the trace function

Because the trace function uses system resource to output the trace information, this lowers the performance of JavaPOS. Use the trace function only when trouble develops on EPSON JavaPOS ADK, to report details of the trouble.

6.2 Usage

The epson.trace.properties file is required before you can run the trace function. The real file and symbolic link of this file are created in the following directory during installation of EPSON JavaPOS ADK, as described in "3.4.3 Installing EPSON JavaPOS ADK and EJEE."

- *<directory where EPSON JavaPOS ADK is installed>/*
  (real file)
- *<directory where Java2 SDK is installed>*/jre/lib
  (symbolic link)

You can use the epson.trace.properties file for the following purposes.

- Enable/disable trace
- Specify the log file name
- Specify the maximum size of the log file

By default, the epson.trace.properties file has the following content, but you can modify it as necessary.

```
# Trace mode.
```

```
# 0 : Not trace.
# 1 : Trace and output to local file.
epson.trace.mode=0

# Log file name.
epson.trace.filename=trace.log

# Log file size.(bytes)
# <= 0 : Not limit file size.
# >  0 : limit file size.
epson.trace.filesize=1048576
```

[Description of the epson.trace.properties file]

- A line beginning with "#" is a comment line.
- epson.trace.mode determines the trace mode. Set it to 0 to disable the trace mode, or set it to 1 to enable the trace mode. By default, the trace mode is disabled.
  (Default: 0(disable))
- Use epson.trace.filename to specify the log file name. The path can be either a full or relative path name. For the relative path, the directory where the application (that uses JavaPOS) starts is the current directory.
  (Default: trace.log)
- Use epson.trace.filesize to specify the maximum size of the log file, in bytes. You can set the size up to 2 GB. If it is set to 0, then no upper limit is imposed on the log file size.
  (Default: 1048576(= 1MB))

The trace function applies to all devices that use the device service of EPSON JavaPOS ADK, and all logs are recorded in the same log file.

6.3 Description of the log file

This section describes the format of the log file used for the trace function. The log file is a text file, and trace information is stored per line in the file. The file has the following format.

TIME LOGICALNAME(ID ACTION) : DESCRIPTION

TIME Indicates time lapsed after the trace function has begun (when the first service is opened), in milliseconds.

LOGICALNAME Indicates the logical device that made an action.

ID A number assigned whenever some action occurs through all services.

ACTION Depending on whether the method or event begins or ends, the following text strings are stored in ACTION. For each ID, there is always a pair of Start and End recorded in the file.

Start	Description
Start	Indicates a method or event has begun.

End	Description
Return	Indicates the method returning a value has ended.
End	Indicates a method that does not return a value, or an event callback has ended.
Error	Indicates an exception or error has occurred when the method was executed.

DESCRIPTION The ACTION of the method or event results in the following information being stored.

ACTION	Description
Start	Indicates the name of the method or event, as well as its content.
Return	Indicates the return value of the method.
End	Usually does not store anything. However, for ErrorEvent, ErrorResponse is stored.

Error Indicates content of a stack trace.

6.4 Example of log file

The following uses the log file of a POS printer as an example (partial).

[POS printer log file example]

```
313880 MyPrinter1(682 Start ) : open("MyPrinter1", jpos.POSPrinter$POSPrinter
Callbacks@1a9b65)
313895 MyPrinter1(682 End)
316057 MyPrinter1(766 Start ) : claim(1000)
317127 MyPrinter1(766 End)
318853 MyPrinter1(809 Start ) : setDeviceEnabled(true)
318865 MyPrinter1(810 Start ) : StatusUpdateEvent
            status - 26
318916 MyPrinter1(809 End)
320606 MyPrinter1(810 End)
319037 MyPrinter1(822 Start ) : getClaimed()
319037 MyPrinter1(822 Return) : true
320607 MyPrinter1(897 Start ) : StatusUpdateEvent
            status - 12
321479 MyPrinter1(897 End)
325146 MyPrinter1(950 End)
325635 MyPrinter1(986 Start ) : printNormal(2, "12345
")
325836 MyPrinter1(986 End)
329568 MyPrinter1(1024 Start ) : setDeviceEnabled(false)
329569 MyPrinter1(1024 End)
331528 MyPrinter1(1067 Start ) : release()
332306 MyPrinter1(1067 End)
333417 MyPrinter1(1110 Start ) : close()
333419 MyPrinter1(1110 End)
```

Appendix-A Device setting

A.1 TM-J2000/TM-J2100/TM-T90/TM-L90

Set the Dip switches of these devices as follows:

1) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	available
2	ON	available
3	OFF	OFF only
4	OFF	OFF only
5	OFF	Note1
6	OFF	Note1
7	ON	Note2
8	OFF	Note2

Dip-SW 2

No.	Setting	
1	OFF	OFF only

Note1 No.5 and 6 of Dip-SW1 set the parity.

Note2 No.7 and 8 of Dip-SW1 set the communication speed.

2) Supported USB I/F, and Ethernet I/F

Dip-SW 1

No.	Setting	
1	OFF	available
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only

Dip-SW 2

No.	Setting	
1	OFF	OFF only

A.2 TM-U210A

Set the dipswitches of this device as follows:

3) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	Note1
6	OFF	Note1
7	OFF	Note2
8	OFF	Recommended

Dip-SW 2

No.	Setting	
1	OFF	OFF only
2	ON	ON only
3	OFF	OFF only
4	OFF	OFF only
5	ON	ON only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only

Note1 No.5 and 6 of Dip-SW1 set the parity.

Note2 No.7 of Dip-SW1 sets the communication speed.

4) Ethernet I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	Recommended

Dip-SW 2

No.	Setting	
1	OFF	OFF only
2	ON	ON only
3	OFF	OFF only
4	OFF	OFF only
5	ON	ON only
6	OFF	OFF only
7	OFF	OFF only
8	ON	ON only

A.3 TM-U230

Set the dipswitches of this device as follows:

1) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	Note1
6	OFF	Note1
7	OFF	Note2
8	OFF	Recommended

Dip-SW 2

No.	Setting	
1	OFF	OFF only
2	ON	ON only
3	OFF	OFF only
4	OFF	OFF only
5	ON	Available
6	OFF	OFF only
7	OFF	OFF only
8	OFF	Available

Note1 No.5 and 6 of Dip-SW1 set the parity.

Note2 No.7 of Dip-SW1 sets the communication speed.

2) Ethernet I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	Recommended

Dip-SW 2

No.	Setting	
1	OFF	OFF only
2	ON	ON only
3	OFF	OFF only
4	ON	ON only
5	ON	Available
6	OFF	OFF only
7	OFF	OFF only
8	OFF	Available

A.4 TM-U950

Set the Dip switches of these devices as follows:

3) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	OFF only
2	OFF	Note 1
3	OFF	Note 1
4	OFF	Note 2
5	OFF	Note 2
6	OFF	Note 3
7	OFF	OFF only
8	OFF	OFF only

Dip-SW 2

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	ON	ON only
4	OFF	Recommended
5	OFF	OFF only
6	ON	ON only
7	OFF	OFF only
8	OFF	OFF only

Note 1 No.2 and 3 of Dip-SW1 set the parity.

Note 2 No.4 and 5 of Dip-SW1 set the communication speed.

Note 3 No.6 of Dip-SW1 sets the connection type of Line Display.

A.5 Other POSPrinter

Set the dipswitches of the devices as follows:

1) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	Note1
6	OFF	Note1
7	OFF	Note2
8	OFF	Note2

Dip-SW 2

No.	Setting	
1	OFF	Recommended
2	OFF	Note3
3	OFF	Available
4	OFF	Available
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only

Note1 No.5 and 6 of Dip-SW1 set the parity.

Note2 No.7 and 8 of Dip-SW1 set the communication speed.

Note3 No.2 of Dip-SW2 sets the connection type of Line Display.

2) Printer that supports USB I/F and Ethernet I/F

Dip-SW 1

No.	Setting	
1	OFF	OFF only
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only

Dip-SW 2

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	Available
4	OFF	Available
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	ON	ON only

A.6 DM-D110/210

Set the dipswitches of this device as follows:

1) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	Note1
4	OFF	Note1
5	OFF	Note2
6	OFF	Note2
7	OFF	Note2
8	OFF	OFF only

Note1 No.3 and 4 of Dip-SW1 set the parity.

Note2 No.5 to 7 of Dip-SW1 set the communication speed.

2) USB I/F

Dip-SW 1

No.	Setting	
1	OFF	OFF only
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	ON	ON only
8	OFF	OFF only

A.7 DM-D500

Set the dipswitches of this device as follows:

1) Serial I/F

Dip-SW 1

No.	Setting	
1	OFF	Recommended
2	OFF	OFF only
3	OFF	OFF only
4	OFF	Note1
5	OFF	Note1
6	OFF	Note2
7	OFF	Note2
8	OFF	Note2

No.	Setting	
1	OFF	OFF only
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only

Note1 No.4 and 5 of Dip-SW1 set the parity.

Note2 No.6 to 8 of Dip-SW1 set the communication speed.

2) USB I/F

Dip-SW 1

No.	Setting	
1	OFF	OFF only
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	ON	ON only

Dip-SW 2

No.	Setting	
1	OFF	OFF only
2	OFF	OFF only
3	OFF	OFF only
4	OFF	OFF only
5	OFF	OFF only
6	OFF	OFF only
7	OFF	OFF only
8	OFF	OFF only