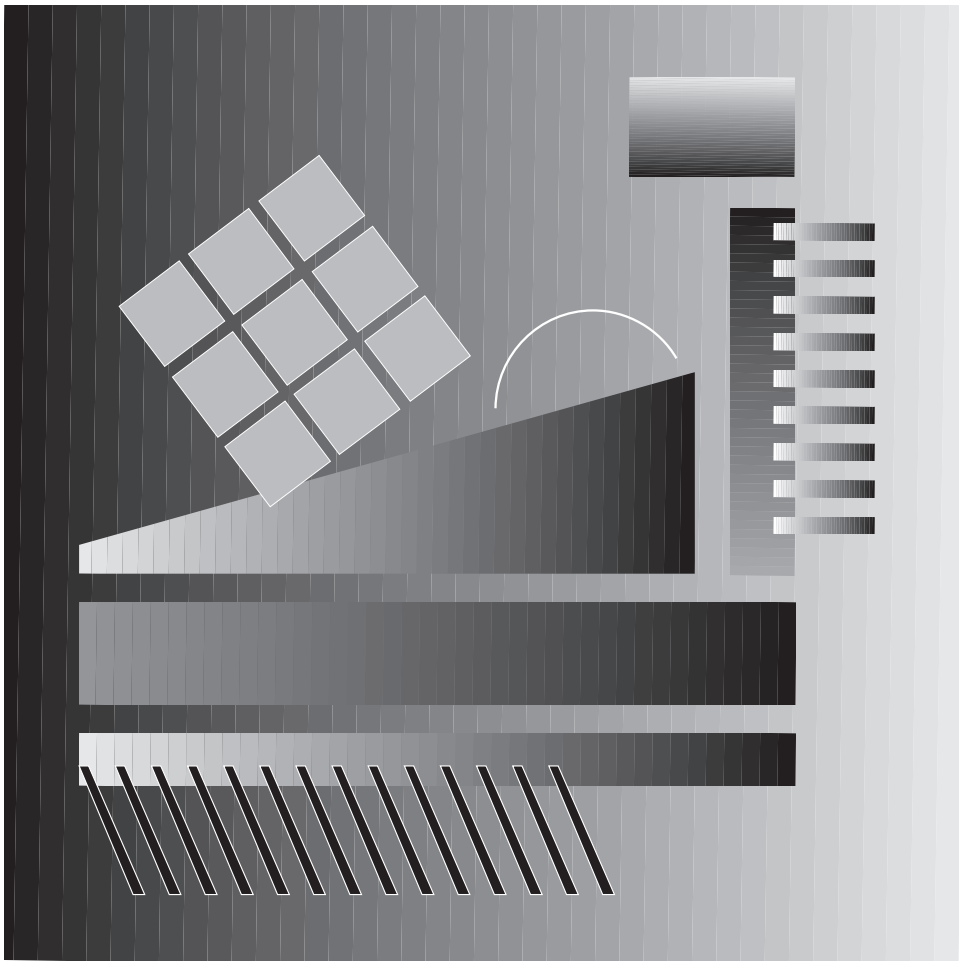
Point of Sale Subsystem

UnifiedPOS User's Guide, Keyboards, and Code Pages



Updated: March 27, 2009

GC30-4078-12

Point of Sale Subsystem

UnifiedPOS User's Guide, Keyboards, and Code Pages

Updated: March 27, 2009

GC30-4078-12

Notice

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 403.

March 2009

This edition applies to Version 1.12.0 of the UnifiedPOS subsystem and to all subsequent releases and modifications until otherwise indicated in new editions.

Retail Store Solutions documentation is available on the IBM Retail Stores Solutions Web site at: http://www.ibm.com/solutions/retail/store/support. Click **Publications**.

A form for reader’s comments is also provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
Retail Store Solutions Information Development
Dept. ZBDA
P.O. Box 12195
Research Triangle Park, NC 27709-9990 USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

About this guide

This guide contains reference information for IBM® Unified Point of Service (UnifiedPOS or UPOS) implementations. This guide also includes information about IBM keyboards and code pages.

Who should read this guide

This guide is intended for use by point-of-sale application developers who need to access IBM point-of-sale hardware using UnifiedPOS.

How this guide is organized

This guide is organized into the following parts:

- Part 1, “User’s guide,” on page 1 introduces UnifiedPOS and describes the installation and device configuration for IBM UnifiedPOS implementation.
- Part 2, “Programming reference,” on page 37 provides the UnifiedPOS properties and methods, and information on resolving problems.
- Part 3, “Keyboards and code pages,” on page 255 provides information about the available keyboards and character sets.

Related publications

This section lists related publications for product components. For information about ordering these publications, contact your IBM authorized dealer or marketing representative.

The latest softcopy version of this guide is available on the IBM Retail Store Solutions web site:http://www.ibm.com/solutions/retail/store/support

- Click **Publications**.
- Look for *Point of Sale Driver Subsystems* under *IBM Store Systems Hardware Publications*.

UPOS publications

Unified Point of Service (UPOS) Retail Peripheral Architecture publications can be found at: http://www.nrf-arts.org/UnifiedPOS/default.htm

Related software

Utility software, LAN drivers, video drivers, and diagnostic software are available. See the latest list on the IBM® Retail Store Solutions Web site at: http://www.ibm.com/solutions/retail/store

Select **Support**, and then click the appropriate hardware or peripheral driver.

Summary of changes

March 2009 - GC30-4078-12
This edition includes updates for UPOS 1.12.0.

Changed or new information is indicated by a revision character (|) in the left margin.

August 2008 - GC30-4078-11
This edition includes UPOS 1.9.6.

April 2008 - GC30-4078-10
This edition includes UPOS 1.9.5.

October 2007 - GC30-4078-09
This edition includes UPOS 1.9.4.

May 2007 - GC30-4078-08
This edition includes UPOS 1.9.3.

September 2006 - GC30-4078-07
This edition includes UPOS 1.9.2.

April 2006 - GC30-4078-06
This edition includes UPOS 1.9.1.

Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373 has been added.

November 2005 - GC30-4078-05
This edition includes UPOS 1.9.

In this edition, Line Scanner and Omni Scanner are added.

Properties and methods tables are updated.

July 2005 - GC30-4078-04
This edition includes new information about UPOS 1.7.5.

January 2005 - Web-only update
This update contains new or additional information related to the IBM Anyplace Kiosk (4836/4838).

December 2004 - GC30-4078-03
This edition contains new or additional information about OLE for Retail Point of Sale (OPOS).

September 2004 - GC30-4078-02
This edition contains new or additional information on device-specific notes, JavaPOS-supported properties and methods, and POS keyboard layouts and scan codes.

May 2004 - GC30-4078-01
This edition includes information about IBM Retail Environment for SuSE Linux (IRES) IBM Retail Environment for SuSE Linux® (IRES).

Part 1. User's guide

Chapter 1. Introduction

UnifiedPOS is an architectural specification developed by the UnifiedPOS technical committee, of which IBM Corporation is a member. The UnifiedPOS architectural specification is for application interfaces to point-of-service (POS) devices that are used in the retail environment. The architectural specification standard is independent of a specific operating system and language, and defines a set of retail device behaviors sufficient to support a range of POS solutions.

IBM UnifiedPOS includes driver packages for both JavaPOS and Object Linking and Embedding (OLE) for Retail Point of Sale (OPOS).

- **JavaPOS** is an implementation of the UnifiedPOS specification written in Java™ for the support of Java applications.
- **OPOS** is an implementation of the UnifiedPOS specification for Windows® systems. The drivers are implemented using COM, and are delivered as ActiveX controls.
- **OPOS Gateway** provides an OPOS interface to the IBM JavaPOS drivers. The OPOS Gateway provides access to functions that are available in JavaPOS but have not been implemented in OPOS, enabling application developers to choose the best level of supported function for OPOS.

System requirements

This section describes the hardware, software, disk space, and memory that are required for the IBM UnifiedPOS 1.12.0 Subsystem.

Hardware environment

This section lists the hardware devices that UnifiedPOS supports.

Point of sale terminals

The following point-of-sale terminals are supported:

Table 1. Point-of-sale terminals supported by UnifiedPOS

Terminal	Models
4674 POS Terminal (Japan only)	All
4694 POS Terminal	2x5, 2x6, 2x7, 3x7
	OPOS only: 0x1, 0x4, Sx1, Sx4, 1x4, 2x4
SurePOS 100 Series	All
SurePOS 300 Series	All
SurePOS 500/600 Series	All
SurePOS 700 Series	All
Kiosk	All
IBM SureOne® POS system	All

RS-485 devices should only be connected to the POS terminal when the terminal is powered off; connecting any RS-485 device to the terminal when the terminal is running could cause damage to the device and/or the terminal and is not supported.

Software environment

The IBM UnifiedPOS system requires the following software environment:

Table 2. Software environment for IBM UPOS

Operating system	One of the following: • Microsoft® POSReady • Microsoft Windows® XP Service Pack 2 • Microsoft Windows Embedded Point of Service (WEPOS) • Microsoft Windows Vista Business • Microsoft Windows Vista Ultimate • SUSE Linux Enterprise Desktop 11 (SLED 11). This is for JavaPOS only.
Java Virtual Machine (JVM)	IBM Java runtime environment (JRE) or software development kit (SDK), version 1.5.0 SR9. To determine JVM version, at the command prompt enter: `java -fullversion`.
Control Objects	OPOS common control objects are required to run the IBM OLE for Retail Subsystem. They can be obtained from http://www.monroecs.com/oposccos.htm or the IBM UPOS installer.

Chapter 2. Installing UnifiedPOS

This chapter describes UnifiedPOS installation for Windows and IBM Retail Environment for SuSE Linux (IRES).

Installation for Windows

You have two options for installing the IBM UnifiedPOS for Retail POS Suite: interactive installation and silent installation.

Interactive installation

1. Run the Setup.exe file and follow the directions on each panel. You might be prompted to install the Microsoft .Net Framework.
2. In the Features dialog (Figure 1), you can choose either of the following features:
   - OPOS Device Support
   - JavaPOS Device Support

   The system unit information will be displayed to the user if the system is recognized.

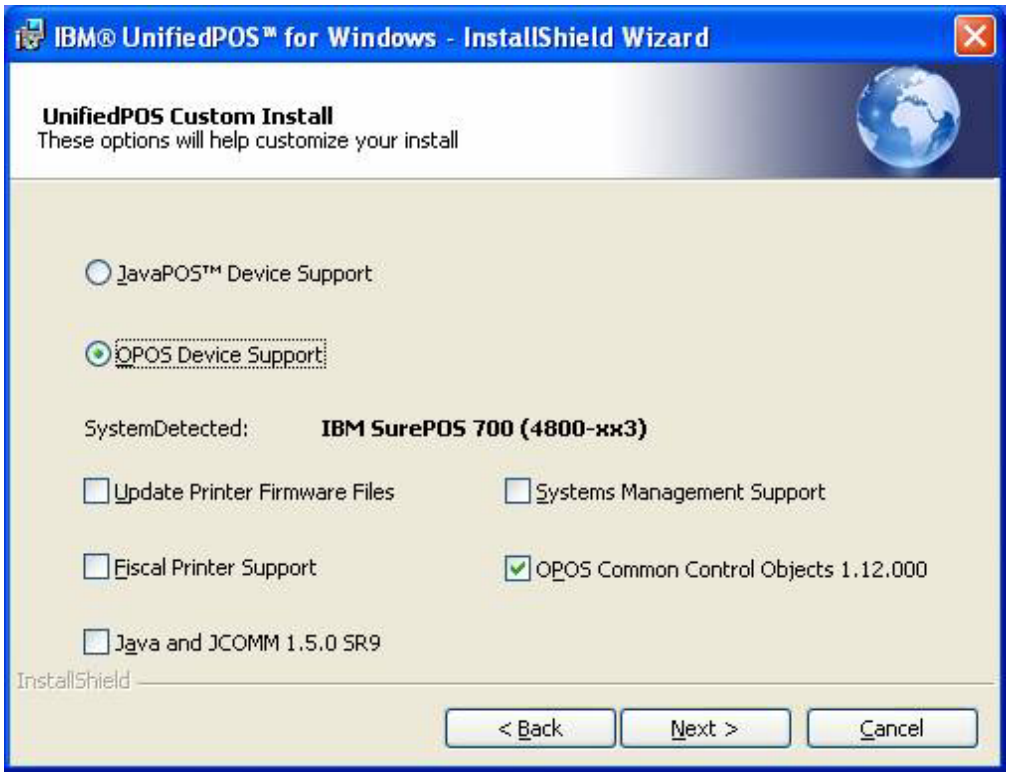


Figure 1. Features dialog

3. After the installation is complete, you must restart your system for the configuration changes to take effect.

Notes:

1. If you select OPOS Common Control Objects during OPOS installation, the installation will override any existing OPOS Common Control Objects on your system.
2. It is the your responsibility to re-register the OPOS Common Control Objects if IBM UnifiedPOS is uninstalled.

Silent installation

IBM UnifiedPOS can be installed and updated silently (unattended) using a response file. The response file, C:\UposSetup.iss is created during the initial installation of the package on a POS system. This response file can then be used to perform an unattended install or update on other systems.

When installing silently on Vista, the user will see both the User Access Control and logo signing messages. To hide these messages, the user must disable both the UAC and the logo signing messages.

To deploy the installation to another system, copy the setup.exe file and the response file to the target system. The response file must be placed in the Windows root directory.

Unattended installation with reboot

To perform an unattended installation or update without a user interface, use the following command:

```
setup.exe /s /v/qn
```

To perform an unattended installation or update with a basic user interface showing progress, errors, and completion, use the following command:

```
setup.exe /s /v/qb
```

To enable logging for the installation, use one of the following commands:

```
setup.exe /s /v"/qn /l*vx C:\msilog.txt"
setup.exe /s /v"/qb /l*vx C:\msilog.txt"
```

Unattended installation without reboot

To prevent the automatic reboot following installation or update, use one of the following commands:

```
setup.exe /s /v"/qn REBOOT=A"
setup.exe /s /v"/qn REBOOT=A /l*vx C:\msilog.txt"
```

Notes:

1. There is no space between **/v"** and **/qn** or between **/v"** and **/qn**.
2. The commands **/qn** and **/qb** are interchangeable in all the examples shown
3. The double quotes are required as shown.
4. The file C:\msilog.txt contains the install log.

Unattended installation flags

In addition the default flags available, the user can also set the following parameters:

USEINI

Allows the user to specify the location of the .iss file to use during silent install. If this parameter is not specified, the default value of [WindowsVolume]UposSetup.iss is used.

SAVEINI
Allows the user to specify where to save the .iss file during installation. If this parameter is not specified, the default value of [WindowsVolume]UposSetup.iss is used.

REMOVEINI
Setting this value to 1 will remove the .iss file from the target machine. Specify the USEINI parameter to remove a file saved in a non-default location.

ONLYINI
Setting this parameter to 1 allows the user to run the UI installation and generate an iss file without actually installing on that system. The installation will exit right after clicking “Install” in the “ReadyToInstall” dialog.

Some examples of these commands are as follows:

```
setup.exe /s /v"/qn USEINI=C:\UposSetup.iss"
setup.exe /v"SAVEINI=C:\UposSetup.iss"
setup.exe /x /v"REMOVEINI=1"
setup.exe /v"ONLYINI=1"
```

Uninstallation

You can uninstall UnifiedPOS from Windows by interactively using the **Add/Remove Programs** option in the Windows Control Panel.

To perform an unattended uninstall, use one of the following commands:

```
setup.exe /s /x /v/qn
setup.exe /x /s /v/qb
```

Note: There is no space between **/v** and **/qn**.

For 4840 systems

Before installing UPOS, in Device Manager under "Multifunction adapters", you will see:

```
IBM 4840-xx3 PCI Multi-I/O Controller
```

No driver files are required or loaded for this device.

After installing JPOS, in Device Manager under "Multifunction adapters", you will see:

```
IBM 4840 System Port
```

This replaces the 4840-xx3 PCI port listed above.

The driver for the IBM4840 System Port is:

```
c:\windows\system32\drivers\aipmtn.sys
```

After uninstalling UPOS (and the IBM 4840 System Port), when you reboot, the hardware manager will display `IBM 4840-xx3 PCI Multi-I/O Controller`. The device manager will prompt you to reload the drivers for this device.

Installation for SUSE Linux Enterprise (SLE)

JavaPOS drivers have been tested with SLED 11.

Visit the following Web sites for more information on Novell Linux products:

- http://www.novell.com/linux/

To obtain information about IBM JavaPOS drivers, submit a TechLine request at the following Web site:

- http://www.ibm.com/solutions/retail/store/support/

Chapter 3. Configuring devices

This chapter describes the configuration process for JavaPOS devices, OPOS devices, and USB access.

JavaPOS configuration

The JavaPOS devices are configured using the jpos.xml file. This section describes how to create a jpos.xml file and provides the information you need to configure certain devices.

Creating a jpos.xml file

Locating sample jpos.xml files

During installation, a sample XML file is installed onto your system. The sample file is named `'jposSample.xml'`.

It can be found in the following folders:

- Linux - /opt/ibm/javapos/docs
- Windows: - c:\pos\ibmjpos\docs

The sample XML files contain JPOS entries for all IBM JavaPOS-supported devices.

To create the jpos.xml file

You can create a new jpos.xml file using one of the following methods.

Creating a jpos.xml file using your favorite text editor: To create a jpos.xml file using your favorite text editor, follow these steps:

1. Create an empty jpos.xml file.
2. Copy the header information from the sample XML file into your new jpos.xml file.
3. Copy only those JposEntries that correspond to devices on the target system from the sample XML file into your jpos.xml file.
4. Change the Logical Name in each JposEntry so that it matches the Logical Name for that device in your JavaPOS application.
5. Save the jpos.xml to the pos\ibmjpos\config directory in Windows, or to the /opt/ibm/javapos/etc directory in Linux. Update the CLASSPATH if jpos.xml file is saved to another location.
6. In your new jpos.xml file, keep only those JposEntries that correspond to devices on the target system.

Creating a jpos.xml file using the JavaPOS Configuration Editor: You can create a jpos.xml file using the JavaPOS Configuration Editor by following these steps:

1. Copy the sample XML file to the jpos.xml file.
2. Change the directory to the location of the jpos.xml file.
3. Open the JavaPOS Configuration Editor by typing:

   ```
   java jpos.config.simple.editor.JposEntryEditor
   ```

   Note: The JavaPOS Configuration Editor can also be accessed from **Programs->IBM JavaPOS->JavaPOS Configuration Editor**.

4. Change the Logical Name in each JposEntry to match the Logical Name for that device in your JavaPOS application.
5. In your new jpos.xml file, keep only those JposEntries that correspond to devices on the target system.

Changing the logical name of JposEntry

Follow these steps to change the logical name of JposEntry:

1. Select a device entry.
2. Click **Edit > Copy**.
3. Enter a new Logical Name in the JposEntry dialog box. Select **OK**.
4. Click **Yes** to delete the old entry.
5. For each device that is not on the target machine, perform the following steps:
   a. Click the JposEntry for that device.
   b. Click **Edit > Delete**.
6. Click **File > Save JposEntryRegistry As...** to save jpos.xml.

Device configuration

The devices are configured based on the entries in the jpos.xml file. Because JposEntries consume resources, it is recommended that jpos.xml contain only those JposEntries that correspond to devices on the target system. For example, if you are working with only EIA-232 devices, the jpos.xml should not include entries for RS-485 or USB devices.

The logical name used for a given device during open must match the LogicalName specified in the JposEntry in the jpos.xml for that device.

The entries in the `jpos xml` file are organized by device type.

EIA-232 device communictaion

Entries for the EIA-232 devices in the `jpos.xml` file include the EIA-232 communication configuration properties. Acceptable values for each EIA-232 communication property are documented in Table 3.

Table 3. EIA-232 device communication properties

JCL property name	Type	Possible values (typical values)	Default Value
portName	String	Valid port name such as: "COM1"	"" (empty string)
baudRate	String	"2400", "4800", "7200", "9600", "14400", "19200", "38400", "57600", "115200"	"9600"
dataBits	String	"4", "5", "6", "7", "8"	"8"
parity	String	"Even", "Odd", "None", "Mark", "Space"	"None"
stopBits	String	"1", "1.5", "2"	"1"
flowControl	String	"Xon/Xoff", "Hardware", "None"	"None"

Notes:

1. The value for the portName depends on the POS system to which the device is attached. All the ports start with the letters "COM"; a number is appended to identify the port, for example: ″COM2″.
2. In Linux the designations COM1 and COM2 are still used instead of the standard Linux terminology.

Device-specific EIA-232 Configuration

Possible values for the device communication properties for specific EIA-232 devices are shown in Table 4. The *dataBits*, *parity* and *stopBits* properties are omitted from the table and from jpos.xml entries.

Notes:

1. Values different than those described here will cause the device to fail.
2. In the following table, `Not required` means that the user does not have to type these values into the xml.
3. For the embedded devices in the 4610 Printer, such as magnetic ink character recognition (MICR) and CheckScanner, the jpos.xml must include the JposEntry for the 4610 Printer and the embedded devices.

Table 4. Specific values for EIA-232 devices in JavaPOS

Device	portName	baudRate	flowControl
CashDrawer 4610-A CashDrawer 4610-B	Port where the printer is attached ″COM1″ (default)	"9600" "19200" "115200" (Must match printer configuration)	Only "Xon/Xoff" or "Hardware" (Must match printer configuration)
CashDrawer 4840-A CashDrawer 4840-B	"COM4" (default)	Not required	Not required
CheckScanner 4610-TI8/TI9	Port where the printer is attached. Default: ″COM1″	"9600" "19200" (Must match printer configuration)	Only "Xon/Xoff" or "Hardware"
Fiscal Printer 4610-Kx3 Fiscal Printer 4610-Kx4 Fiscal Printer 4610-Kx5 Fiscal Printer 4610-Gx3 Fiscal Printer 4610-Gx4 Fiscal Printer 4610-Gx5	Port where the printer is attached Default: ″COM1″	"9600" "19200" "28800" (Windows only)	Only "None"
Fiscal Printer 4610-Sx6	Port where the printer is attached Default: ″COM1″	"115200"	Only "None"
Line Display 4840 APA	Default: "COM4"	"9600"	"Hardware"
Line Display 4810/4840/SureOne VFD	Default: "COM4"	Not required	Not required
MICR 4610 TI2/4/8/9	Port where the printer is attached Default: ″COM1″	"9600" "19200" (Must match printer configuration)	Only "Xon/Xoff" or "Hardware"
MSR 4820/4836/4838/4840 ISO MSR 4830/4840 JUCC	Default: "COM3"	"19200"	Not required

Table 4. Specific values for EIA-232 devices in JavaPOS (continued)

Device	portName	baudRate	flowControl
POSPrinter 4610 TI1/2/3/4/8/9 TM6/7, TF6/7	Port where the printer is attached Default: ″COM1″	"9600" "19200" (Must match printer configuration)	Only "Xon/Xoff" or "Hardware"
POSPrinter SureOne	Only "COM3"	"9600"	Only "Xon/Xoff"
IBM Anyplace Kiosk Line Scanner	Default: "COM4"	"19200"	Not required
IBM Anyplace Kiosk Omni Scanner	Default: "COM4"	"9600"	Not required
Tone Indicator 4610 TM/TF 6/7	Port where the printer is attached Default: ″COM1″	"9600" "19200" (Must match printer configuration)	Only "Xon/Xoff" or "Hardware"

POS Control Center utility

The POS Control Center displays information for devices found on the system and from device entries from the jpos.xml file. For multiple, identical devices, such as cash drawer and displays, the deviceNumber property becomes important. For a given device, the device number displayed in the Device Information section of the POS Control Center must match the deviceNumber in the JposEntry.

To access the IBM POS Control Center:

- In Windows, click **Programs** > **IBM JavaPOS** > **POS Control Center**.
- In Linux, enter `POSControlCenter`.
- From a command prompt in either operating system, enter:
  ```
  java com.ibm.jpos.tools.sdicc.ControlCenterApp
  ```

SurePOS 300-33x, JavaPOS second cash drawer support and NVRAM detection

The 4810-x3x POS hardware has an embedded setting which shows that there is a second cashdrawer, as well as nonvolatile random access memory (NVRAM) through JavaPOS. Neither the second cashdrawer nor the NVRAM are valid I/O devices on the SurePOS 300-33x POS hardware. Although these devices are present on the JavaPOS ControlCenter, they are not configurable, and must not be configured in the jpos.xml file as valid devices.

You should be aware that when using the IRES POS client configuration utility, these devices (second cashdrawer and NVRAM) will be generated as valid entries. However, this is an invalid configuration and must not be used as configurable devices.

IBM SurePoint Display (4820) Touch Support

The touch support is not provided for the following:

1. IBM SurePoint Display (4820) RS485 device on Windows Vista
2. IBM Retail Environment for SuSE Linux Version 2

JavaPOS Auto-configuration (Linux only)

JavaPOS Auto-configuration can detect most POS devices that are physically connected to the system. It will create a new JCL entry with the default values for each device it detects. However, some configuration must be done manually. For example, the following devices must be manually configured.

- All EIA-232 devices except for SurePOS 500 cash drawers
- Cash drawers on IBM 4610 POS printers

See “Manual device configuration” for more information.

Any device settings can be changed by editing the specific device properties file. For example, scanner and scale settings need to be manually configured.

Note: The Auto-configuration utility is supported through the IRES POS Client Configuration utility. For details on the use of this utility, refer to the *IBM Retail Environment for SUSE Linux Version 2 Developer’s Guide* at http://www.ibm.com/solutions/retail/store/support,

- Under *Operating Systems*, click **IRES**.
- Click on the latest version of the **Developers Guide**.
- To open the pdf, click on the language you want.

Configuring the JCL

There are three modes of JCL operation for device configuration:

1. Auto-configuration only

   This mode creates a JCL table with entries created through auto-configuration only. To enable this mode, copy the file located in the `/opt/ibm/javapos/etc/jpos/res/jpos.properties.autocfg` directory to the `/opt/ibm/javapos/etc/jpos/res/jpos.properties` directory.
2. XML and auto-configuration

   This mode creates a JCL table with entries created from both the jpos.xml file and auto-configuration. To enable this mode copy the file located in the `/opt/ibm/javapos/etc/jpos/res/jpos.properties.xmlautocfg` to the directory `opt/ibm/javapos/etc/jpos/res/jpos.properties` directory.
3. XML only

   This mode creates a JCL table with entries created from the jpos.xml file only. To enable this mode copy the file located in the `/opt/ibm/javapos/etc/jpos/res/jpos.properties.xmlcfg` directory to the`opt/ibm/javapos/etc/jpos/res/jpos.properties` directory.

Manual device configuration

When using Auto-configuration (JCL operation mode (1) or (2) above), there is an additional configuration file for every device. The device properties files are located in the `/opt/ibm/javapos/etc` directory. Table 5 shows the name of the file associated with each device type.

Table 5. Auto-configuration properties files

Device	Properties file name
CashDrawer	posj.autoconfig.CashDrawer.properties
CheckScanner	posj.autoconfig.CheckScanner.properties
FiscalPrinter	posj.autoconfig.FiscalPrinter.properties
HardTotals	posj.autoconfig.HardTotals.properties

Table 5. Auto-configuration properties files (continued)

Device	Properties file name
Keylock	posj.autoconfig.Keylock.properties
LineDisplay	posj.autoconfig.LineDisplay.properties
MICR	posj.autoconfig.MICR.properties
MSR	posj.autoconfig.MSR.properties
MotionSensor	posj.autoconfig.MotionSensor.properties
POSKeyboard	posj.autoconfig.POSKeyboard.properties
POSPrinter	posj.autoconfig.POSPrinter.properties
Scale	posj.autoconfig.Scale.properties
Scanner	posj.autoconfig.Scanner.properties
ToneIndicator	posj.autoconfig.ToneIndicator.properties

Note: All of these additional configuration files that support EIA-232 configuration include a section related to EIA-232 settings. The configuration for an EIA-232 device must contain at least the following properties:

- deviceBus.1=RS232
- portName.1=COM1
- baudRate.1=9600
- flowControl.1=Xon/Xoff

By default, all EIA-232 properties are commented out in these configuration files. To enable an EIA-232 device, its corresponding file must be edited to include the lines that configure the settings for the device.

Auto-configuration properties files format

For some properties it may be necessary to specify a different value than the one assigned by default. Every property can be set in 2 different formats:

Default format:

Used to set a value to a property that will be used for all devices of the same class. Its format is:

```
property = value
```

Number-specific format:

Used to set a property value only for a specific device. Other devices of the type will keep the default values.

```
property.number = value
```

where:

- property = Any of the device properties available for the device
- number = A numeric sequence of the devices of the same category, starting at 1, which must match to the one assigned to the LogicalName property, so this property can be associated to the correct device.
- value = Any valid value for the property

The following examples show how to set the value for a property in the 2 formats:

- To enable the property stripTransitDashes for all the MICR devices autodetected in the system the default format is used:

  ```
  com.ibm.jpos.sdi.config.MICR.stripTransitDashes=true
  ```

- To enable the property stripTransitDashes for only the first MICR device the number-specific format is used:

  ```
  com.ibm.jpos.sdi.config.MICR.stripTransitDashes.1=true
  com.ibm.jpos.sdi.config.MICR.stripTransitDashes=false
  ```

LogicalName property considerations:

The logicalName property must be unique to a particular device so it can only be set using the number-specific property format. If no Number-specific logicalName property is defined for a detected device, Auto-configuration automatically generates a logicalName for the device by combining the device category and the enumeration number of that device.

For example, the logicalName of the first CashDrawer enumerated would be CashDrawer1 and the logicalName of the third LineDisplay enumerated would be LineDisplay3.

The logicalName can be modified by editing the `posj.autoconfig.<device>.properties` file for the specific device.

EIA-232 device configuration

EIA-232-connected devices are difficult to auto-detect, so the specific device properties file must be edited to specify some or all of the EIA-232 settings. These settings must be specified in number-specific format.

At a minimum, the EIA-232 portName and deviceBus must be set. The following example shows how the settings should look to configure an EIA-232 IBM 4610 printer:

```
logicalName.1=POSPrinter1
deviceBus.1=RS232
portName.1=COM1
baudRate.1=19200
flowControl.1=Xon/Xoff
```

Note: A cash drawer connected to a SurePOS 500 is the only auto-detected EIA-232 device; the EIA-232 settings should not be specified.

Configuring cash drawers on IBM 4610 POSPrinters

Cash drawers attached to the IBM 4610 POS Printer cannot be automatically detected, even when connected to the printer.

To enable the cash drawers, modify the file posj.autoconfig.POSPrinter.properties and enable the cash drawer properties. For example:

```
autoconfig.hasCashDrawerA.1=true
autoconfig.hasCashDrawerB.1=true
```

MICR exception tables configuration

Auto-configuration handles exception table entries using a list of entries saved in a file, one entry per line. The following example shows how this file might look:

```
com.ibm.jpos.sdi.config.MICR.exceptionTable1= B778899001D154R
com.ibm.jpos.sdi.config.MICR.exceptionTable2= B667788990D153R
com.ibm.jpos.sdi.config.MICR.exceptionTable3= P123456780AAAAAXXSSS
```

The posj.autoconfig.MICR.properties file should be edited to specify the name of the MICR exceptionTable file in the `autoconfig.exceptionTableFile property`.

```
autoconfig.exceptionTableFile=<micrentries.txt file>
```

Note: Refer to “MICR exception tables” on page 112 for information about the formats.

Displaying Auto-configuration detected devices

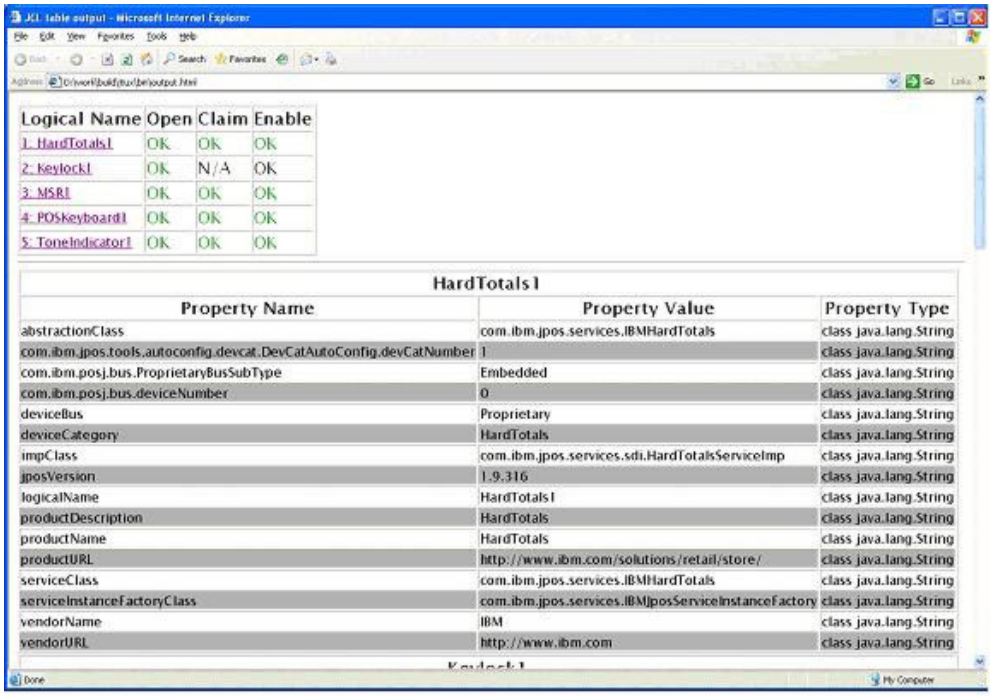
The auto-config.tool shows, in html format, all the devices that were detected by Auto-configuration, along with the assigned logicalName and properties. At a command prompt, type:

```
java com.ibm.jpos.tools.autoconfig.tool.ReadJCL <output html file>
```

where:

output.html is the file that will contain the information generated by auto-configuration, listing all the detected devices and its properties.

This tool also performs basic operations (open, claim, enable) on each detected device and store the result in the <output.html>. The following image is an example of the output when HardTotal and POS keyboard device and sub devices are detected.



Logical Name	Open	Claim	Enable
1: HardTotals1	OK	OK	OK
2: Keylock1	OK	N/A	OK
3: MSR1	OK	OK	OK
4: POSKeyboard1	OK	OK	OK
5: ToneIndicator1	OK	OK	OK

HardTotals1

Property Name	Property Value	Property Type
abstractionClass	com.ibm.jpos.services.IBMHardTotals	class java.lang.String
com.ibm.jpos.tools.autoconfig.devcat.DevCatAutoConfig.devCatNumber	1	class java.lang.String
com.ibm.posj.bus.ProprietaryBusSubType	Embedded	class java.lang.String
com.ibm.posj.bus.deviceNumber	0	class java.lang.String
deviceBus	Proprietary	class java.lang.String
deviceCategory	HardTotals	class java.lang.String
impClass	com.ibm.jpos.services.sdi.HardTotalsServiceImp	class java.lang.String
jposVersion	1.9.316	class java.lang.String
logicalName	HardTotals1	class java.lang.String
productDescription	HardTotals	class java.lang.String
productName	HardTotals	class java.lang.String
productURL	http://www.ibm.com/solutions/retail/store/	class java.lang.String
serviceClass	com.ibm.jpos.services.IBMHardTotals	class java.lang.String
serviceInstanceFactoryClass	com.ibm.jpos.services.IBMJposServiceInstanceFactory	class java.lang.String
vendorName	IBM	class java.lang.String
vendorURL	http://www.ibm.com	class java.lang.String

Figure 2. Result of running com.ibm.jpos.tools.autoconfig.tool.ReadJCL

Auto-configuration supported devices

The JavaPOS Auto-configuration code detects a subset of devices supported by the JavaPOS drivers. The following tables present the devices that can be detected.

Cash drawers:

Table 6. List of supported cash drawers

Device	Connectivity
SurePOS 500/600 family built-in cash drawer	EIA-232
SurePOS 700 family built-in cash drawer	USB/Integrated

Table 6. List of supported cash drawers (continued)

Device	Connectivity
4694 family built-in cash drawer	RS-485
SurePOS 300 family	Integrated
SureOne family built-in cash drawer	Integrated

CheckScanners:

Table 7. List of supported CheckScanners

Device	Connectivity
4610-TI8 CheckScanner	RS-485/USB
4610-TI9 CheckScanner	RS-485/USB

Fiscal printers:

Table 8. List of supported Fiscal Printers

Device	Connectivity
4610 Kxx/Gxx Fiscal Printer family	RS-485/USB/Rs232
4610 Sx6 Fiscal Printer family	USB/Rs232
4679 3xx Fiscal Printer family	RS-485

HardTotals:

Table 9. List of supported HardTotals

Device	Connectivity
4694 NVRAM	Integrated
SurePOS 700 Series NVRAM	Integrated/USB
SureOne built-in NVRAM	Integrated

Keylocks:

Table 10. List of supported keylocks

Device	Connectivity
Alphanumeric POS Keyboard keylock	PS/2 / RS-485 / USB
Alphanumeric POS Keyboard keylock with mouse pointer, MSR	PS/2
50-key Keyboard keylock	RS-485/USB
133-key Keyboard keylock	RS-485/USB
4820 keylock	RS-485/USB
SureOne keylock	Integrated
Alphanumeric POS Korea Keyboard keylock	PS/2 / RS-485 / USB

LineDisplays:

Table 11. List of supported LineDisplays

Device	Connectivity
SBCS 40-character VFD II	RS-485/USB
SBCS Two-sided VFD II	RS-485/USB
SBCS 40-character VFD II (Japan)/4674	RS-485
SBCS Character Graphic (C/G) APA	RS-485/USB
SBCS 40-character LCD	RS-485/USB
SBCS 50-key keyboard LCD	RS-485/USB

MICRs:

Table 12. List of supported MICRs

Device	Connectivity
4610 Printer family MICR	RS-485/USB

MSRs:

Table 13. List of supported MSRs

Device	Connectivity
Alphanumeric POS Keyboard MSR	PS/2 / RS-485 / USB
Alphanumeric POS Keyboard keylock with mouse pointer, MSR	PS/2
50-key keyboard MSR	RS-485/USB
133-key keyboard MSR	RS-485/USB
4820 MSR	RS-485/USB
SureOne built-in	PS/2
CANPOS keyboard(firmware version 1.2.3 or higher)	PS/2

MotionSensors:

Table 14. List of supported MotionSensors

Device	Connectivity
SurePOS 500/600	Integrated
Kiosk	Integrated

POS keyboards:

Table 15. List of supported POS keyboards

Device	Connectivity
Alphanumeric POS keyboard	PS/2 / RS-485 / USB
Alphanumeric POS keyboard with mouse pointer	PS/2
50-key keyboard	USB/RS-485

Table 15. List of supported POS keyboards (continued)

Device	Connectivity
133-key keyboard	USB/RS-485
32-key keypad	PS/2 / RS-485 / USB
CANPOS keyboard	PS/2
POS Keyboard V	RS-485/USB
Retail POS keyboard	RS-485/USB
SureONE built-inKeyboard	PS/2

POSPrinters:

Table 16. List of supported POSPrinters

Device	Connectivity
4610 Printer family	RS-485/USB
4689 Printer family	RS-485/USB

Scales:

Table 17. List of supported Scales

Device	Connectivity
IBM 4696 compatible scanner/scale	RS-485/USB
IBM 4698 compatible scanner/scale	RS-485/USB
OEM Scales (must conform to the IBM USB OEM POS Device Specification)	USB

Scanners:

Table 18. List of supported scanners

Device	Connectivity
IBM 1520-compatible hand-held	RS-485
IBM 4696-compatible scanner	RS-485/USB
IBM 4697-compatible scanner	RS-485/USB
IBM 4698-compatible scanner	RS-485/USB
IBM HHBCR-compatible	RS-485
IBM HHBCR2-compatible	RS-485
OEM Scanners	USB
IBM 4685 Scanner	RS-485/USB
Kiosk Scanner	Integrated

ToneIndicators:

Table 19. List of supported ToneIndicators

Device	Connectivity
Alphanumeric POS Keyboard tone	PS/2 / RS-485 / USB

Table 19. List of supported ToneIndicators (continued)

Device	Connectivity
Alphanumeric POS Keyboard with mouse pointer tone	PS/2
50-key Keyboard tone	RS485/USB
133-key Keyboard tone	RS485/USB
4820 tone	PS/2 / RS-485 / USB
PC POS Keyboard	RS485/USB
4610 SST Tx6/Tx7	RS-485/USB

Note: Not all OEM devices have been tested and verified.

USB device access

Microsoft Windows

With one exception, IBM peripherals are USB 1.1 compliant, which can cause communications problems when connected to a USB 2.0 controller. It is strongly recommended that USB 2.0 be disabled in the terminal BIOS settings.

The exception to this is the IBM 4610 2NR/2CR printer, which is a 2.0 compliant device and does not require that USB 2.0 be disabled.

Linux

The IBM JavaPOS drivers use the javax.usb subsystem to access the USB POS devices connected to the system. The `javax.usb` subsystem on Linux uses the USB `filesystem, usbfs,` to access the USB devices directly. The permissions on this `filesystem` default to access only by the root user. This security model prevents unauthorized users from accessing USB devices connected to your system. However, if the JavaPOS application (for example, JVM) is running as a non-root user, the usbfs permissions must be changed to allow non-root access. To do this you must change the mount parameters for usbfs. You can edit the `/etc/fstab` file to modify the mount parameters for usbfs (fourth field) or you can manually remount usbfs, specifying the required parameters on the *mount* command.

The most secure way to accomplish non-root access to the USB device nodes is to change the ownership of the USB device nodes to the userid which is running the JavaPOS application. To do this, mount usbfs with the parameter, `devuid=n`, where *n* is the numeric ID of the JavaPOS application's user. For example, if the JavaPOS application is running as user, `javaposapp`, and `javaposapp` has a User ID (uid) of 1000, then specify `devuid=1000` when mounting usbfs. The user running the JavaPOS application must be considered a privileged user because this user can directly access all USB devices. Users who do not need to run the JavaPOS application or the JavaPOS drivers should use a different login.

A less secure way to accomplish non-root access to the USBdevice nodes is to create a special group for the users who run JavaPOS applications. Two parameters must be used in mounting usbfs: `devgid=n`, where *n* is the numeric group id for the JavaPOS application group, and `devmode=0664`, which changes the permissions of the USB device nodes to read/write by owner and group. The group given permission to access the USB devices should be considered a privileged group and handled the same as a privileged user. Only those users that need to run JavaPOS applications and drivers should be made part of that group.

See the Linux man pages for `useradd, usermod, groupadd`, and *groupmod* for information about creating and modifying users and groups.

OPOS configuration

IBM OPOS drivers must be configured to access the point-of-sale devices you have attached to your system. To locate the Configuration Utility, select **Programs > IBM OLE for Retail Point of Sale > Configuration Utility**. OPOS provides defaults for all resources associated with supported devices. This section discusses how the application can configure a device to use a value different from the assigned default and how to specify some resource values.

Using the OPOS Configuration utility

The OPOS configuration window has a separate tab for each of the four POS Terminal types supported by the IBM OPOS package (see Figure 3).

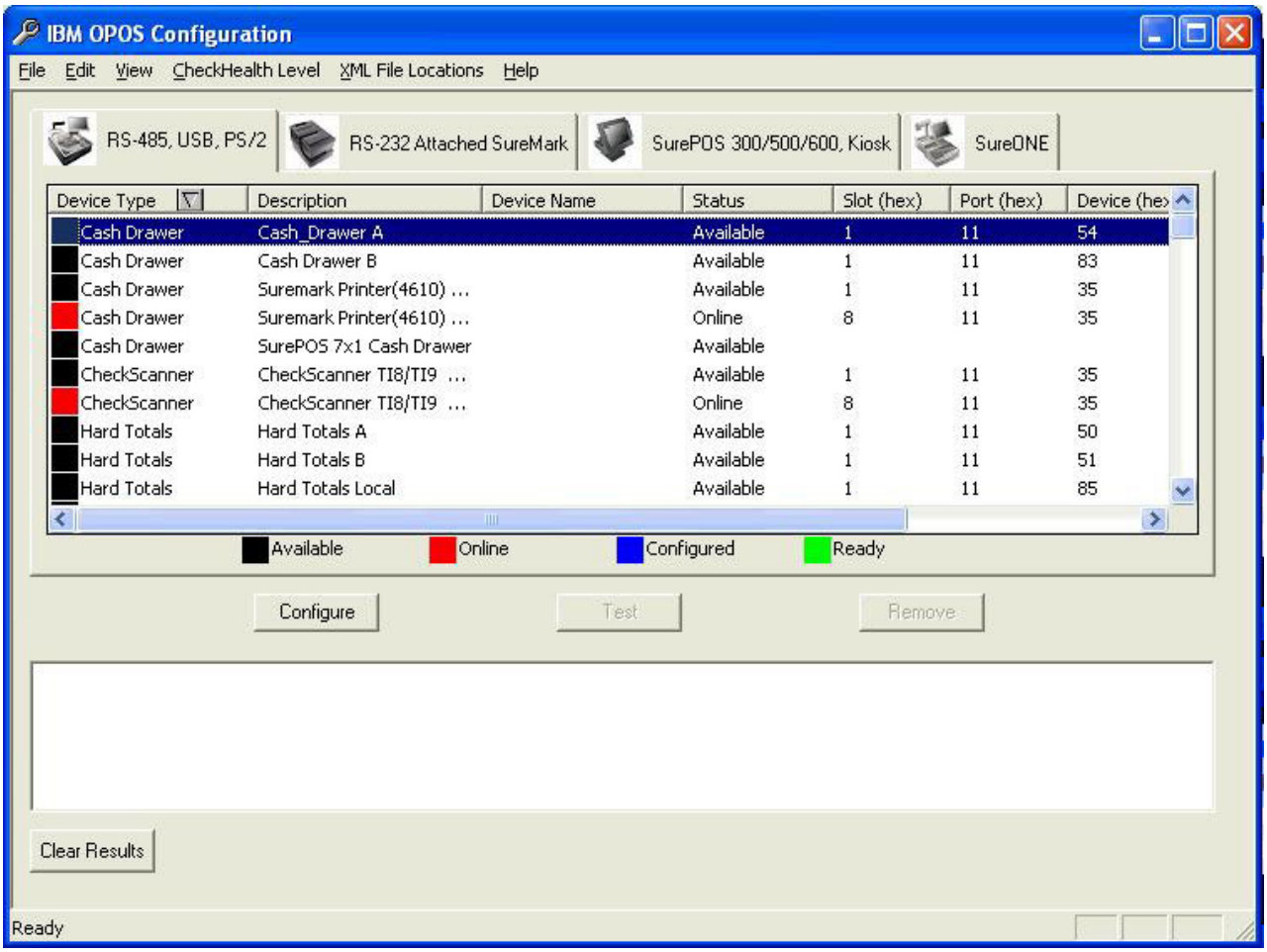


Figure 3. OPOS configuration window

Each tab contains a list of devices that are supported on the POS Terminal. Each device entry contains descriptive information and the device's current status. The status shows what level of configuration is completed using a status description. Table 20 on page 23 explains the meaning of each status type.

Table 20. Device status types

Status	Color	Description
Available	Black	The device is supported on the selected POS Terminal. It can only be used to add a new configuration. Its settings will not be modified. No device name is given.
Online	Red	The device is supported on the selected POS Terminal, and is detected to be online and available on the current system. It can only be used to add a new configuration. Its settings will not be modified. No device name is given.
Configured	Blue	The device is supported on the selected POS Terminal and has a configuration entry in the registry. It can be modified, tested for connectivity, or removed.
Ready	Green	The device is supported on the selected POS Terminal and is ready to use. It is online and has a configuration entry in the registry. It can be modified, tested for connectivity, or removed.

RS-485, PS/2, and USB devices are detected (and listed as Online) automatically. The system tone and HardTotals are always listed as Online. Other devices can be listed as Online after testing.

Navigation

For finding specific device entries, you can sort a column by clicking its heading. The column is sorted in ascending order first; click again to sort in descending order. The current sort order is indicated by the up/down arrow in the column heading.

Note: The order is based on ASCII values. (For example, a baud rate of 19200 is listed before 9600 when the column is sorted in ascending order.) Only one column can be sorted at a time.

Configuring devices

To configure a device, perform the following steps:

1. Select an entry in the device list. The **Configure** button is enabled.
2. Click the **Configure** button to display the configuration dialog for the selected device. The dialog differs depending on the POS Terminal type and the device type. Figure 4 on page 24 shows the configuration dialog for an RS-485 or USB device; Figure 5 on page 25 shows the configuration dialog for an EIA-232 device.
3. Enter a name for the device.
4. For USB, RS-485, and System devices, select the correct slot and port for the device (see Figure 4 on page 24). Slot can have the following values:

Slot	Description
0	System devices (keyboard, NVRAM)
1	Most RS-485 devices
2	Some RS-485 devices (4674 devices)
8	USB devices

Port can have the following values:

Port (hex)	Description
0	System devices (keyboard)
11	Most RS-485 devices
22	RS-485 devices (46x3 Model 2)

Note: The device number cannot be modified because that number is specific to the type of device, and changing it could change the device type.

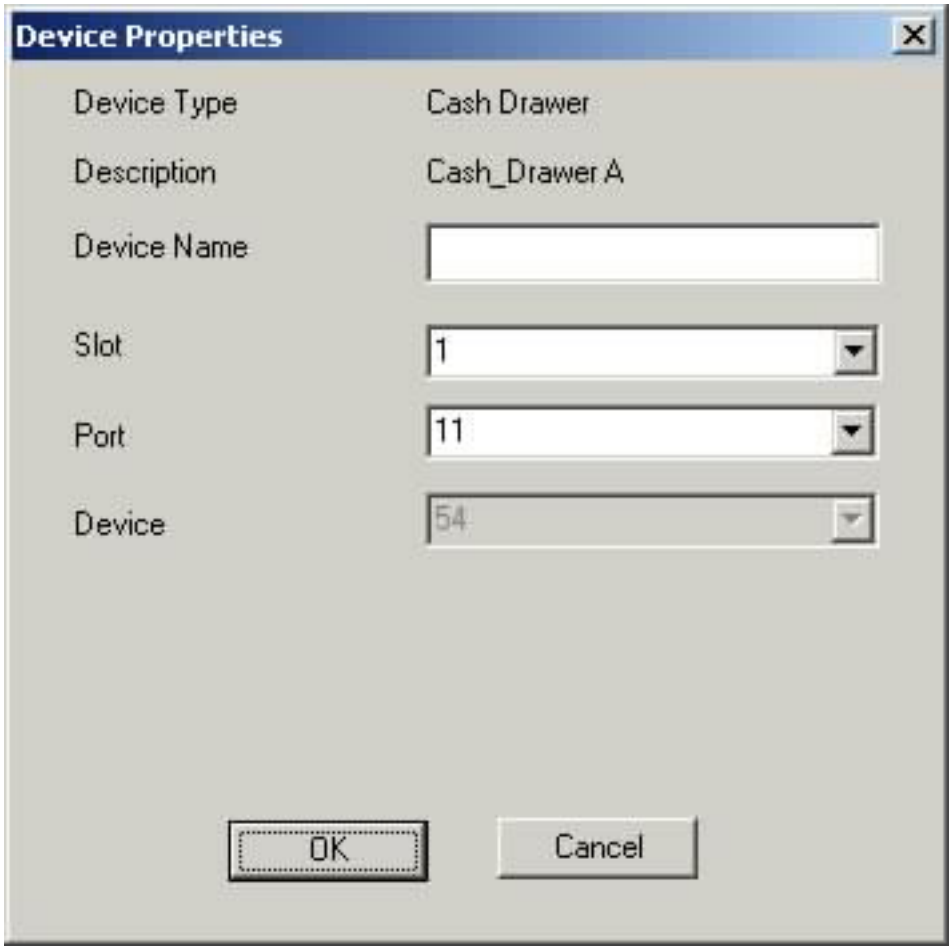


Figure 4. Device properties dialog (RS-485/USB)

For EIA-232, select the correct Com Port, Baud Rate, and Flow Control for the device (See Figure 5 on page 25).

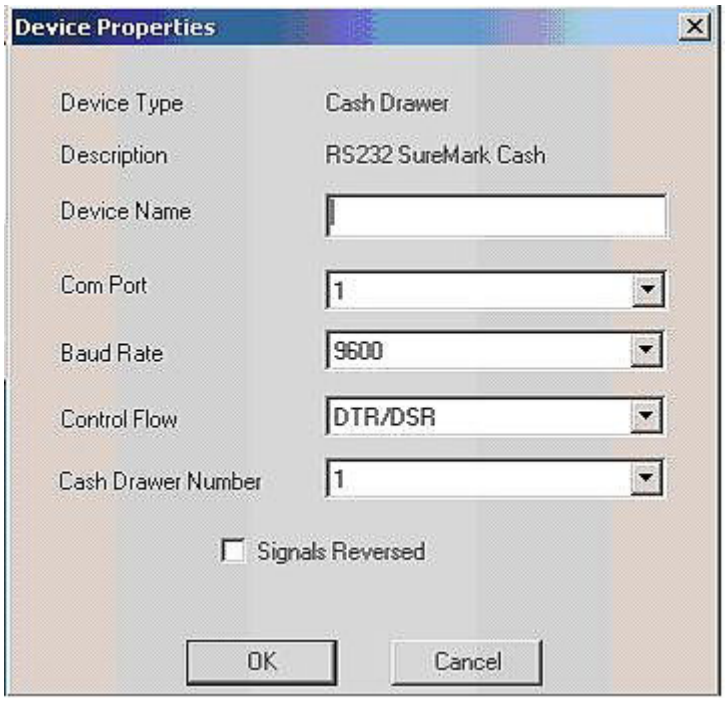


Figure 5. Device properties dialog (EIA-232)

5. Click **OK**. The entry is added to the device configuration list.

Notes:

1. Devices that are not attached can still be configured.
2. For RS-485 and USB devices, Slot and Port should not have to be changed.
3. RS232 devices will not show as Online in the OPOS configuration tool until they are configured.
4. For more information about Slot, Port, and Device numbers, see the *Point of Sale Subsystem: Programming Reference and User's Guide*.

Testing connectivity

Selecting a Configured or Ready device enables the **Test** and **Remove** buttons, as well as the **Configure** button. Clicking on the **Test** button tests the connectivity and configuration of the device. While the device does not need to be attached to configure or remove a device configuration entry, it must be attached for a successful test. The **Test** button uses OPOS to open, claim when necessary, and enable the device. It then performs a CheckHealth method. The CheckHealth level is set from the menu.

If all of these steps are completed successfully, then the test is considered successful. If the device configuration entry is not yet marked as Ready, it is updated as such.

The list box at the bottom of the window is updated with the test results (see Figure 6 on page 26).

Figure 6. Results of connectivity test

Removing devices

You can delete entries during configuration. The results are displayed in the list box at the bottom.

Deploying OPOS device configurations

You can use the configuration utility to import and export configuration information to and from an OPOS Device Registry (ODR) file. These options are available on the **File** menu.

To export configuration information, perform the following steps:

1. Select the devices (on all the tabs) that you want to export. Use the **Shift** and **Ctrl** keys to select more than one. (See Figure 7 on page 27.)

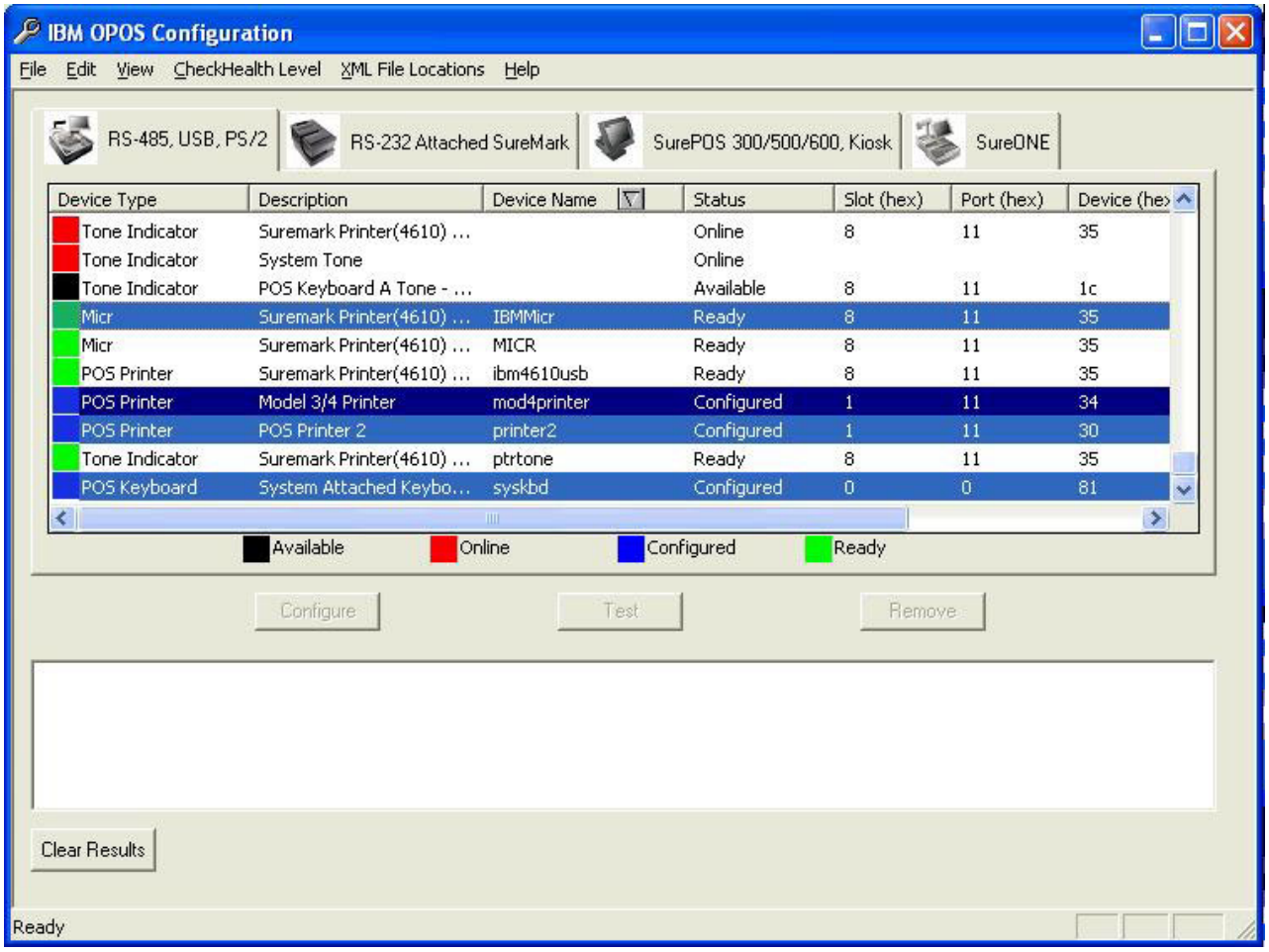


Figure 7. Selecting devices to export

2. Select **File > Export**. The configuration information is exported.

Only configured entries with device names are exported. This enables you to create files for different configurations, or for deploying new configurations to stores.

To import configuration information, perform the following steps:

1. Select **File > Import**. A file dialog window is displayed.
2. Select an ODR file and click **Open**. The configuration information is imported and the device listing and registry information are updated.

IBM SurePoint Display(4820) Touch Support

Touch support is not provided for the IBM SurePoint Display (4820) RS485 device on Windows Vista.

Chapter 4. Problem determination

This chapter describes the logging files that record device events. These events allow you to determine and resolve problems.

JavaPOS problems

Tracing

UPOS 1.9.1 or later tracing can be configured using the UPOS Support Tool. See Appendix B, “UnifiedPOS support tool,” on page 395 for more information.

Note: The tracing facility described here does not work when you’re using JavaPOSAutoConfig from IRES V2. Refer to the *IBM Retail Environment for SUSE LINUX Version 2 Developer’s Guide*, form number GC30-9723 for more information.

The POS Control Center provides a facility to enable device tracing to collect trace information. This facility is described in the following list:

1. Start POS Control Center in one of two ways:
   a. **Programs > IBM JavaPOS > POS Control Center**,
   b. Command line: **java com.ibm.jpos.tools.sdicc.ControlCenterApp**
2. Select **File > Properties**.
3. Select the **jutil** tab.
4. Modify the applicable properties; press enter key after each change:

```
com.ibm.jutil.tracing.TracerOutputTo=File (default).
com.ibm.jutil.tracing.TurnOnAllNamedTraces=OFF (default), to enable set to ON.
com.ibm.jutil.tracing.TracerOutputFileLocation=~/.ibmjpos (default)
com.ibm.jutil.tracing.TracerOutFileName=traceroutput.txt (default)
com.ibm.jutil.tracing.TracerLevel=Maximum (default).
com.ibm.jutil.tracing.TimeStampFormat = DATE (default)
      Use TimeStampFormat to give format in milliseconds [1140564390381]
      or in Date plus hour [yyyy.MM.dd H:mm:ss:SSS]=[2006.02.24 12:42:17:039]
      give TimeStampingFormat property one of the following values :
      for milliseconds: MSECS
      for Date: DATE  ( default value )
com.ibm.jutil.tracing.TracerOutputFileMaxSize=10000000(default) contents the
      maximum number of bytes that a file can handle and store
      --default is 10000000 ( 10 Mbytes ).
com.ibm.jutil.tracing.TracerOutputFileMaxFiles=10 (default) contents the maximum
        number of files where the data traced is stored  --default 10 file.
```

   Example: If a trace contains 500 bytes, the TracerOutputFileMaxSize property is set to 100, and the TracerOutputMaxFiles is set to 4, then four files of 100 bytes each will be created:

   - traceroutput1.txt
   - traceroutput2.txt
   - traceroutput3.txt
   - traceroutput4.txt

   The file traceroutput1.txt contains the end of the tracing, while traceroutput4.txt contains the second 100 bytes of the trace. The first 100 bytes of the trace are lost.

Note: In Windows, the property com.ibm.jutil.tracing.TracerOutputFileLocation must not end with a single backslash. A single backslash indicates that the line continues. Use two backslashes.

Refer to the POS Central Center manual for details about properties. To access the manual, go to **Programs > IBM JavaPOS > POS Control Center > Help > Manual**.

Optionally, the tracer output file can be sent to a logging facility, which can rotate the trace files. The following steps describe how to send the tracer output file to logging:

1. In jutil.properties set the value of the property com.ibm.jutil.tracing.TracerOutputTo to **Logging**. This sends all the trace messages through logging.
2. In log4jConfig.properties, modify the following properties as needed:

   ```
   log4j.appender.rootAppender.File =<home>/.ibmjpos/logging/root.log (default)
   log4j.appender.rootAppender.MaxFileSize=100000 (default)
   # Keep one backup file
   log4j.appender.rootAppender.MaxBackupIndex=1
   ```

   If MaxbackupIndex is greater than 1, then the log file is rotated.

Tracing for Javax.usb for Windows

Release 1.9.2 for UPOS introduced new tracing capabilities, requiring version 1.0.2 or later of jsr80_ri.jar. There are two layers:

- Java and DLL layer tracing
- kernel driver tracing

Java and DLL layer tracing

Tracing is controlled by javaxusb.properties using the following properties:

- The WindowsUsbTracer property specifies the tracer to use. In this example, tracing is directed to the console (default) by "WindowsUsbTracerStdOut".

  ```
  com.ibm.jusb.os.windows.WindowsUsbTracer=com.ibm.jusb.os.windows.util.
  WindowsUsbTracerStdOut
  ```

- The nativeTraceLevel property sets the debug level for the DLL tracing.

  ```
  com.ibm.jusb.os.windows.WindowsUsbServices.nativetracelevel=TRACE_CRITICAL
  ```

- The current Level property sets the debug level for the javax.usb for Windows layer. Setting this property to TRACE_OFF disables javax.usb for Windows tracing.

  ```
  com.ibm.jusb.util.WindowsUsbTracer.currentLevel=TRACE_OFF
  ```

The following lines show part of a console trace.

```
NATIVE:[TopologyMonitor]-->
NATIVE:[EnumerateJxDevices]Dev path
    \\?\usb#vid_04b3&pid_4526#no_serial_number#{136e983a-096c-
49bb-a6c6-e608a0a0cdbc}
NATIVE:[JxGetDescriptor]getting javaxusb descriptor type: 1. Pass 1
NATIVE:[JxGetDescriptor]getting javaxusb descriptor. Pass 2 with size: 18
NATIVE:[EnumerateJxDevices]Device descriptor return length:18
NATIVE:[JxGetDescriptor]getting javaxusb descriptor type: 3. Pass 1
NATIVE:[JxGetDescriptor]getting javaxusb descriptor. Pass 2 with size: 34
NATIVE:[EnumerateJxDevices]Total JavaxUsb Devices found: 1
```

Note that *NATIVE* identifies DLL tracing, and *STATIC* identifies static methods.

JavaPOS tracing support: In order to use the JavaPOS tracing capabilities, the provided JUtilUsbTracer must be specified in javaxusb.properties:

```
com.ibm.jusb.os.windows.WindowsUsbTracer=com.ibm.jutil.tracing.javaxusb.
JUtilUsbTracer
```

Here is an example:

```
...
</com.ibm.posj.bus.hid.javaxusb.factory.ignoreInterface>
[1153764821690][POSJ][PosSystemManager]-->startPopulators()
[1153764821690][POSJ][PosSystemManager]-->startPopulators()
[1153764821841][JavaxUsb][NATIVE][EnumerateJxDevices]Dev path
\\?\usb#vid_04b3&pid4526#no_serial_number#{136e983a-096c-49bb-a6c6-e608a0a0cdbc}
[1153764821841][JavaxUsb][NATIVE][JxGetDescriptor]getting javaxusb descriptor
    type: 1. Pass 1
[1153764821861][JavaxUsb][NATIVE][JxGetDescriptor]getting javaxusb descriptor.
    Pass 2 with size: 18
[1153764821871][JavaxUsb][NATIVE][EnumerateJxDevices]Device descriptor return
    length:18
```

In this example Javaxusb tracer is identified as [JavaxUsb]. [NATIVE] is tracing generated by the DLL, and [STATIC] is tracing generated by static methods from the Java layer. Anu other tracers come from the Java layer.

The advantage of this feature, apart from integrating JavaPOS-javaxusb tracing, is that all JavaPOS tracing capabilities are incorporated into javaxusb. These include output to file, logging, TCP/IP tracing server, and so on.

Note that in this mode you can filter javaxusb tracing by adding the following line in jutil.properties:

```
com.ibm.jutil.tracing.Tracer.JavaxUsb = OFF
```

Kernel driver tracing

In order to debug the kernel, programs like DebugView from Sysinternals may be used, or use MS Debugger from another computer connected to the serial port.

DebugView requires no special settings to view the trace. If you use it, you might see DLL tracing as well, depending on the DLL tracing configuration of javaxusb.properties.

Tracing for Javax.usb for Linux

The Linux implementation of javax.usb has JNI tracing only. To change the settings for JNI tracing, edit the javaxusb.properties file. The available settings are:

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.tracing = true`

  A value of false disables tracing entirely; a value of true enables some amount of tracing.
- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_output = 2`

  This defines where the JNI tracing output is sent. The default is stderr file mode. Note that if append mode is used, the trace file will grow ever-larger every time javax.usb is run; its size should be managed. In file mode, the file size is not managed or limited by javax.usb.
  - 1 - stdout
  - 2 - stderr
  - 3 - file (truncate mode)
  - 4 - file (append mode)
- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_filename =`

This is the filename to trace to, if trace_output is set to file tracing. There is no default. It must be set if trace_output is set to file tracing.

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_level = 0`

  This sets the tracing level. Higher levels mean more tracing. This level applies to ALL tracers. See each tracer for their levels.

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_default = true`

  This enables or disables default tracing. This is the most used tracer. The trace_data must also be enabled. The levels for this tracer are:
  - 0 - CRITICAL
  - 1 - ERROR
  - 2 - INFO
  - 3 - FUNCTION
  - 4 - DEBUG
  - 5 - OTHER

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_hotplug = true`

  This enables or disables hotplug tracing. The trace_data must also be enabled. These are the levels for this tracer:
  - 0 - CRITICAL
  - 1 - ERROR
  - 2 - CHANGE
  - 3 - DEVICE
  - 4 - OTHER

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_xfer = true`

  This enables or disables xfer tracing. The trace_data must also be enabled. These are the levels for this tracer.
  - 0 - CRITICAL
  - 1 - ERROR
  - 2 - REQUEST
  - 3 - METADATA
  - 4 - DATA
  - 5 - OTHER

- `#com.ibm.jusb.os.linux.LinuxUsbServices.JNI.trace_urb = false`

  This enables or disables urb tracing. The trace_data must also be enabled.

  Note: DATA-level tracing will generate a LOT of output.
  These are the levels for this tracer.
  - 2 - METADATA
  - 3 - DATA

The default settings are tracing enabled at level 0. The level should be increased to a desired level. Also, the output can be changed from stdout to a file or stderr. Finally, as the default is not to trace actual URB data, since it is a lot of trace info, if URB data should be traced the trace_urb must be changed to "true" (and uncommented).

OPOS problems

Tracing

For IBM OPOS device drivers, the following steps are used to gather trace information.

1. Using a command line prompt, type in AIPTRCCFG. The application will appear.

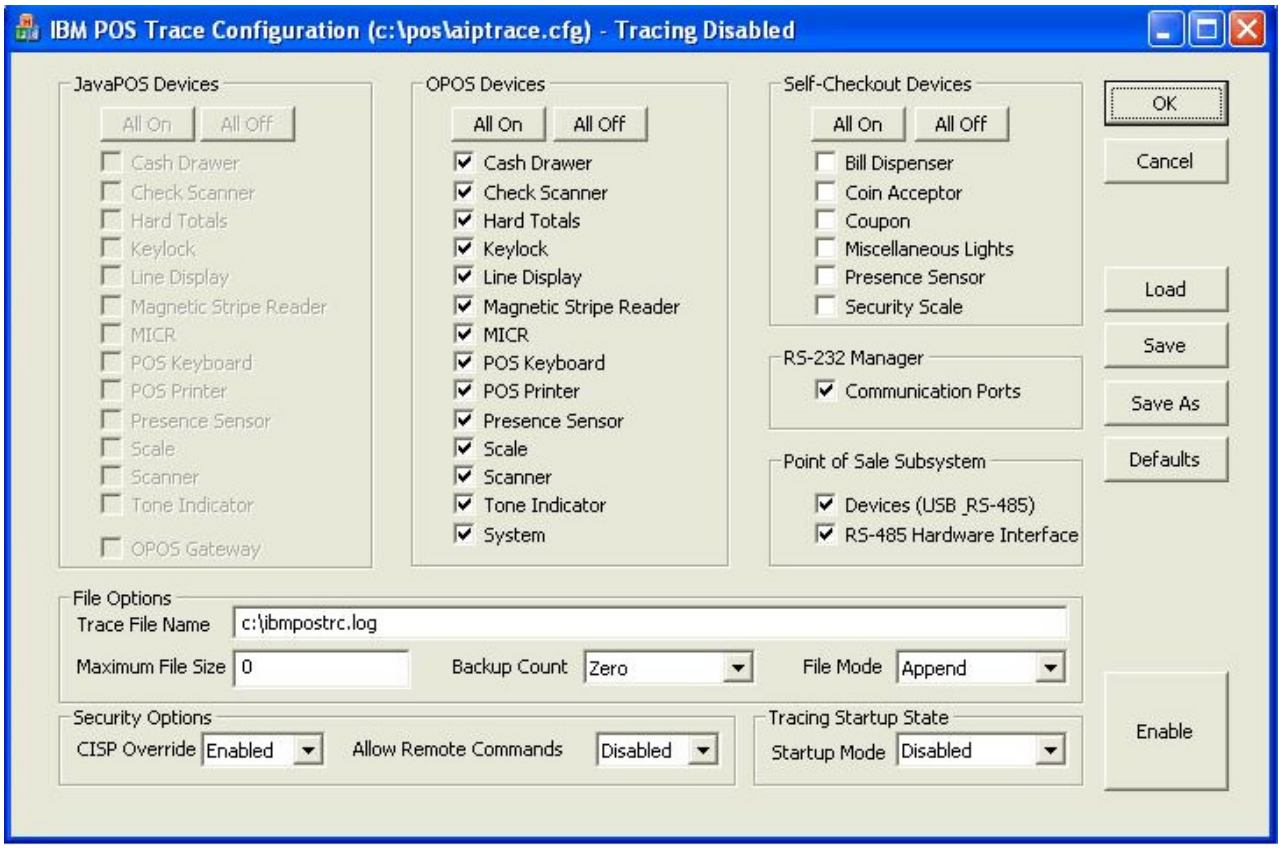


Figure 8. POS tracing facility

2. Choose items requested for tracing.

   When diagnosing printer problems, it is normally recommended to enable tracing for all logical printer devices (cash drawer, check scanner, MICR, POS Printer, Tone Indicator).

   When diagnosing OPOS problems remember to check Communications Ports checkbox (for EIA-232 devices) or Devices checkbox (for USB & RS4-485 devices).

   The Maximum File Size is measured in bytes; 0 implies no maximum.

3. When trace configuration is complete, click **Save** and then click **Enable**.

OPOS Gateway

Trace activation

For OPOS Gateway, you can activate tracing for all devices or for particular devices.

To activate trace for all devices:

1. Go to **Control Panel** -> **System** -> **Advanced (Tab)** -> **Environment Variables** and add the following Environment Variables as SYSTEM Variables:
   - AIP_OPOS_TRACE=ON
   - AIP_OPOS_TRACE_ALL=ON

To activate trace for one or more particular devices:

1. Go to **Control Panel** -> **System** -> **Advanced (Tab)** -> **Environment Variables** and add/modify following Environment Variables as SYSTEM Variables:
   - AIP_OPOS_TRACE=OFF
   - AIP_OPOS_TRACE_ALL=OFF
2. At Windows Registry, locate:
   `HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS Key`
3. Add a value of type DWORD called TraceFlags.
4. Set TraceFlags to a value of 1 up to 4.
   Where value 1 relates to the highest tracing level such as errors and warnings and value 4 relates to the lowest tracing level which shows all method calls.
5. In addition, locate the device key for which the trace is desired:

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\[DeviceCategory]\[DeviceName]key
```

   Where:
   - [DeviceCategory] applies to the Device Category name. For example: POSPrinter
   - [DeviceName] applies to the specific Device name. For example: MyPOSPrinter2
6. Add a value of type DWORD called TraceFlags
7. Set TraceFlags to a value of 1 up to 4. Where value 1 relates to the highest tracing level such as errors and warnings and value 4 relates to the lowest tracing level which shows all method calls.

Trace files location

OPOS GW Trace Files are located at POS\Log directory, the files generated are:

- Oposgw.POSService.Out
  which contains OPOS GW Service related information
- Oposgw.COM[DeviceCategory].Out
  which contains OPOS GW COMProxy layer tracing for a certain DeviceCategory
- Oposgw.so.[DeviceCategory].Out
  which contains OPOS GW ServiceObjects layer tracing for a certain DeviceCategory

Notes:

1. When OPOS Gateway Trace has been activated, the JavaPOS Tracing Files will also be located at POS/Log and these file names will be javapos_oposgw[Number].txt
2. When using a non-administrator account which has no write access to POS\Log, the Oposgw.COM[DeviceCategory].Out files will be located in [Drive]:\Documents and Settings\[UserName]\.ibmjpos.

   Where:

   - [Drive] relates to the letter where Windows is running. For example: C.
     [UserName] relates to the User Account Name.

Getting help

Support Web site

The IBM Retail Store Solutions Web site contains the latest version of the IBM OPOS software as well as fixes to known problems, hints, and tips for using the software. The URL for our Web site is: `http://www.ibm.com/solutions/retail/store/support`

Your first stop for help should be the IBM Retail Store Solutions *Knowledgebase*. It is filled with the latest tips, hints and FAQs on our product lines. It is indexed and you can search on keywords such as **OPOS and printer** to find all related articles. At the bottom of the home page you will see a link to the Knowledgebase. The address is:

`http://www2.clearlake.ibm.com/store/support/html/knowledgebase.html`

Your next stop is to check for updated driver modules. Go to the registration page for the POS drivers you are using and look for the **module update** link. Click it and all changed modules, along with information about where to use them will be listed.

If you have general pre-sale or usage questions about our drivers not answered in the publications and are an IBM Business Partner, you can submit questions to the Partnerline team from our Knowledgebase Web page.

Reporting problems

To report problems, visit the following URL:

`http://www.ibm.com/solutions/retail/store/support/guide`

Part 2. Programming reference

Chapter 5. Common settings and configuration options

This chapter describes implementation and support of UPOS. For each device type, there is a list of the specific devices that are supported and a list of the UPOS properties, methods, and events that are supported. For some devices, there is additional information specific to the JavaPOS and OPOS implementations including additional configuration options as well as DirectIO documentation.

The following properties are used only in OPOS:

- BinaryConversion
- OpenResult
- ResultCode
- ResultCodeExtended

Some properties have different names in OPOS. Table 21 lists the UnifiedPOS and OPOS names for these properties.

Table 21. Alternate property names used in OPOS

UnifiedPOS name	OPOS name
DeviceControlDescription	ControlObjectDescription
DeviceControlVersion	ControlObjectVersion
DeviceServiceDescription	ServiceObjectDescription
DeviceServiceVersion	ServiceObjectVersion
PhysicalDeviceDescription	DeviceDescription
PhysicalDeviceName	DeviceName
claim	ClaimDevice
release	ReleaseDevice

JavaPOS

Configuration options

baudRate

```
<prop name="baudRate" type="String" value="9600"/>
```

Determines the baud rate to use for the EIA-232 device. The value to use depends on the device. Some hardware devices cannot have their flow control value changed.

Refer to the hardware manual for the device to determine the correct value.

deviceBus

```
<prop name="deviceBus" type="String" value="RS232"/>
```

The deviceBus property determines how the device is physically attached to the POS terminal.

It can be one of the following values:

- Proprietary

- RS232
- RS485
- HID

flowControl

```
<prop name="flowControl" type="String" value="Xon/Xoff"/>
```

Determines the flow control used for EIA-232 devices. The value to use depends on the device. Some hardware devices cannot have their flow control value changed.

It can be one of the following values:
- Xon/Xoff
- None
- Hardware

impClass

<prop name="impClass" type="String" value="com.ibm.jpos.services.sdi.ToneIndicatorServiceImp"/>

The impClass specifies the name of the Java class that is responsible for implementing the UPOS specification for a particular device. It uses the Java class specified in the abstractionClass entry to communicate with the physical hardware.

The abstractionClass and impClass are a matched pair of configuration entries that together determine how an application using JavaPOS communicates with a specific device.

Current XML entries should have an impClass entry that contains one of the following values:

```
com.ibm.jpos.services.sdi.CashDrawerServiceImp
com.ibm.jpos.services.sdi.CashDrawer4610ServiceImp
com.ibm.jpos.services.sdi.CheckScannerServiceImp
com.ibm.jpos.services.sdi.FiscalPrinterServiceImp
com.ibm.jpos.services.sdi.HardTotalsServiceImp
com.ibm.jpos.services.poss.IBMHardTotalsImp
com.ibm.jpos.services.sdi.KeylockServiceImp
com.ibm.jpos.services.sdi.LineDisplayAPAServiceImp
com.ibm.jpos.services.sdi.LineDisplayVFDServiceImp
com.ibm.jpos.services.poss.LineDisplayAnop0Imp
com.ibm.jpos.services.poss.LineDisplayShopperImp
com.ibm.jpos.services.sdi.LineDisplayServiceImp
com.ibm.jpos.services.sdi.MICRServiceImp
com.ibm.jpos.services.poss.IBMMICRImp
com.ibm.jpos.services.sdi.MotionSensorServiceImp
com.ibm.jpos.services.sdi.MSRServiceImp
com.ibm.jpos.services.sdi.POSKeyboardServiceImp
com.ibm.jpos.services.sdi.IBM4610PrinterServiceImp
com.ibm.jpos.services.sdi.IBMSureonePrinterServiceImp
com.ibm.jpos.services.sdi.IBM4689PrinterServiceImp
com.ibm.jpos.services.poss.POSPrinter4Imp
com.ibm.jpos.services.sdi.ScaleServiceImp
com.ibm.jpos.services.sdi.ScannerServiceImp
com.ibm.jpos.services.sdi.ToneIndicatorServiceImp
```

You should not need to manually change this value.

portName

```
<prop name="portName" type="String" value="COM1"/>
```

Determines the port to use for EIA-232 devices.

Valid port names are determined by how many ports the POS terminal physically has.

Port names are the same format on Windows and Linux (COMx).

ProprietaryBusSubType

```
<prop name="com.ibm.posj.bus.ProprietaryBusSubType" type="String" value="Embedded"/>
```

For system devices (those have a deviceBus value of Proprietary), this determines whether the device is connected via PS/2 (keylock, tone indicator, MSR, POS Keyboard) or is physically built into the terminal (hard totals, motion sensor, keylock, cash drawer).

It can be one of the following values:
- Embedded
- PosKbd

sioDeviceNumber

```
<prop name="com.ibm.posj.bus.rs485.sioDeviceNumber" type="String" value="0x54"/>
```

A unique number that indicates the type of device. For valid values, refer to the *IBM Point of Sale Subsystem Programming Reference and User's Guide*.

Since RS-485 devices are detected automatically by the driver you should not need to manually change this value.

sioPortNumber

```
<prop name="com.ibm.posj.bus.rs485.sioPortNumber" type="String" value="0x11"/>
```

Determines the logical port number for RS-485 peripheral devices.

Valid values are:
- 0x11 - RS485 devices attached to the primary controller
- 0x22 - RS485 devices attached to the secondary controller

Since RS-485 devices are detected automatically by the driver you should not need to manually change this value.

sioSlotNumber

```
<prop name="com.ibm.posj.bus.rs485.sioSlotNumber" type="String" value="0x01"/>
```

Determines the logical slot number for RS-485 peripheral devices.
- 1 - RS-485 devices attached via the ISA or PCI bus
- 5 - RS-485 devices attached via the micro-channel bus

Since RS-485 devices are detected automatically by the driver you should not need to manually change this value.

USB firmware update files for JPOS

All IBM USB peripheral devices contain firmware to control the USB bus, in addition to the firmware that controls the device itself.

Depending on the device, the firmware for the USB can be packaged in a number of different ways:

1. Separate from the main device firmware (IE, 4610 Tx3/4/5/6/7)

2. Contains the USB bus firmware embedded in the main device firmware (IE 4610 Tx8/9)
3. Contains the main device firmware embedded with the USB bus firmware (IE, USB 2x20 display

The JavaPOS driver will automatically update the firmware, if necessary, when the terminal first boots. It will not update the firmware if a device is hot-plugged while the terminal is running.

New firmware files may be downloaded from the IBM website and placed into the relevant folder on the terminal.

Windows: `\pos\ibmjpos\res\flash\usb`

Linux: `/opt/ibm/javapos/flash/usb`

EIA-232 PosPrinter device detection

When you are using a logical device on an EIA-232 attached printer, you must ensure that the jpos.xml file contains an entry for the device you want to use (for example, MICR) and an entry for the printer itself.

If your jpos.xml file does not contain an entry for the printer then you will be unable to use your logical device.

OPOS

Accessing multiple devices

One physical device (such as a SureMark printer) can contain multiple logical OPOS devices (such as a printer, tone indicator, MICR and cash drawer). If you are accessing multiple logical devices in one physical device at the same time (as many POS applications do), then access all of the logical devices from the same executable program. You cannot use different executable programs to control each single logical device.

Configuration options

OPOS configuration settings are stored in the registry per the Unified POS specification.

The settings in Table 22 on page 49 apply to PS/2, RS485 and USB devices. Used in combination they will uniquely identify a peripheral device.

Because the devices are detected automatically by the driver you should not need to change any of the values.

The settings in Table 23 on page 49 apply to EIA-232 devices. Devices attached via EIA-232 cannot be automatically detected by the driver and will have to be configured manually.

Table 22. Service Object settings for non EIA-232 devices

Keyword	Type	Description	OPOS Gateway Support
slotNumber	String	Logical slot (0-8) to which the device is connected. Typical values are: 0 - System devices 1 - RS485 devices attached via the ISA or PCI bus 5 - RS485 devices attached via the micro-channel bus 8 - USB devices	No
portNumber	String	Logical port to which the device is connected. Typical values are: 0 - System devices X'11' - RS485 devices attached to the primary controller X'22' - RS485 devices attached to the secondary controller	No
deviceNumber	String	Unique number that indicates the type of device. For valid values, refer to *IBM Point of Sale Subsystem Programming Reference and User's Guide*.	No

Table 23. Service Object settings for EIA-232 devices

Keyword	Type	Description	OPOS Gateway Support
ComPort	String	Serial port that the device is attached to. Valid values are determined by the operating system.	Yes
BaudRate	String	Baud rate used for serial port. Valid values are determined by the specific peripheral device. Typical values are 9600, 19200 and 115200.	Yes
ControlFlow	String	Control flow for serial device. Valid values are DTR/DSR or XON/XOFF.	Yes

USB firmware update files for OPOS

All IBM USB peripheral devices contain firmware to control the USB bus, in addition to the firmware that controls the device itself.

Depending on the device, the firmware for the USB can be packaged in a number of different ways:

1. Separate from the main device firmware (IE, 4610 Tx3/4/5/6/7)
2. Contains the USB bus firmware embedded in the main device firmware (IE 4610 Tx8/9)
3. Contains the main device firmware embedded with the USB bus firmware (IE, USB 2x20 display

The OPOS driver will automatically update the firmware, if necessary, when the terminal first boots. It will not update the firmware if a device is hot-plugged while the terminal is running.

New firmware files may be downloaded from the IBM website and placed into the \POS\USB folder on the terminal. The status of the USB firmware can be checked by viewing `\POS\LOG\AIPFLASH.LOG`

Modifying device behavior (USB, RS-485 and PS/2 devices)

For USB, RS-485, and PS/2 devices, additional device configuration can be done through the resource file, `aipsys.res`. Use of this file is documented in *Point of Sale Subsystem: Programming Reference and User's Guide (SC30-3560)*. For general information about using aipsys.res, see Chapter 3, *Customizing the IBM Point Of Sale Subsystem*. For device-specific resources that can be specified in aipsys.res, see Chapter 21, *Resource Sets*.

Notes:

1. Resource names and values are case sensitive.
2. Resource names appear in the documentation as PosNxxxXxxx. However, when specifying resources names in the aipsys.res file, the PosN prefix must be removed.
3. Defined macros for resource values are documented as PosXXXX_XXXX. However, when using these macros in the aipsys.res file, the Pos prefix must be removed.
4. Changes to the aipsys.res file will not take effect until the system is rebooted.
5. On Windows, aipsys.res is located in the default install directory, C:\POS.
6. On Linux systems, aipsys.res is located in the /etc directory.

Ignoring non-IBM USB devices

In some situations it might be necessary to configure the OPOS driver to ignore USB devices manufactured by other vendors.

This can be accomplished by creating a new value in the registry using the information below.

Table 24. Registry value information

Location	HKLM\Software\IBM\Point of Sale Subsystem
Name	DisableNonIBMUSB
Type	String

Table 24. Registry value information (continued)

Value	TRUE

OPOS Gateway

The OPOS Gateway provides an OPOS interface to the IBM JavaPOS drivers, and supports all the function provided by JavaPOS.

Devices that use the OPOS Gateway are configured via OPOSConfig. Although basic registry settings are created to conform to the UPOS specification, configuration items for each device are stored in the JPOS.XML file.

Applications that use the OPOS Gateway are still required to use control objects.

DirectIO calls

In the case of DirectIO, special handling is required when converting the OPOS DirectIO BSTR* parameter to the JavaPOS DirectIO object parameter.

For specific DirectIO methods that require object parameters, the string data must be passed in as an XML representation of the JavaPOS object. The gateway converts the XML string to a JavaPOS object during the DirectIO call.

A COM interface called IDirectIOXMLConverter has been created to facilitate support for DirectIO calls requiring BSTR * to Java object conversion.

Table 25 lists the DirectIO calls that are supported in OPOS Gateway.

To use it, follow these steps:

1. Install the UPOS package containing the OPOSSDI Gateway components.
2. Add a reference for IDirectIOXMLConverter to your project.

Table 25. Devices Supported

UPOS devices that support JavaPOS DirectIO	XML object required?	JavaPOS Object Names
Check Scanner	N	n/a
Fiscal Printer	Y	FiscalInformation
Line Display	N	n/a
MSR	N	n/a
POS Keyboard	N	n/a
POS Printer	Y	DirectIOFontInfo DBCSFontInfo PrintGetDirectIOFontInfo

The public classes in the IDirectIOXMLConverter object are:

- DBCSFontInfo Properties
  - filename as String
- DBCSFontInfo Methods
  - GetDBCSFontInfo
  - GetDBCSFontInfoObject
  - GetDBCSFontInfoXML
- DirectIOFontInfo Properties
  - Station as Long
  - FontNumber as Long
  - FileName as String
  - Len as Long
- DirectIOFontInfo Methods
  - GetDirectIOFontInfo
  - GetDirectIOFontInfoObject
  - GetDirectIOFontInfoXML
- FiscalInformation Properties
  - FiscalCountry as Long
  - FiscalPowerInterrupted as Boolean
  - FiscalVersion as Long
- FiscalInformation Methods
  - GetFiscalInformation
  - GetFiscalInformationObject
  - GetFiscalInformationXML
  - GetByteArray
  - GetByteArrayXML
- PrintGetDirectIOFontInfo Properties
  - receiptCharacterSet(0 to 3) as Long
  - slipCharacterSet(0 to 1) as Long
  - DBCSCodePage as Byte
  - matrixImpactCodePage as Byte
  - matrixUDThermalCodePage1 (0 to 1) as Byte
  - matrixUDThermalCodePage2 (0 to 1) as Byte
  - matrixUDThermalCodePage3 (0 to 1) as Byte
  - matrixUDThermalCodePage4 (0 to 1) as Byte
  - matrixUDImpactCodePage1 (0 to 1) as Byte
  - matrixUDImpactCodePage2 (0 to 1) as Byte
  - DBCSAddressVector1 as Long
  - DBCSAddressVector2 as Long
  - DBCSAddressVector3 as Long
- PrintGetDirectIOFontInfo Methods
  - GetPrintGetDirectIOFontInfo
  - GetPrintGetDirectIOFontInfoObject
  - GetPrintGetDirectIOFontInfoXML

DBCSFontInfo

- **Function GetDBCSFontInfo(XML As String, FileName As String) As String**
  - extracts the FileName field from a DBCSFontInfo object encoded in an XML string
  - returns the string `Success` or a string with the reason if it fails
  - not needed by the application but is provided for convenience
- **Function GetDBCSFontInfoObject(XML As String, obj As DBCSFontInfo) As String**
  - converts an XML string containing an encoded DBCSFontInfo object into a DBCSFontInfo object
  - returns the string `Success` or a string with the reason if it fails
  - not needed by the application but is provided for convenience
- **Function GetDBCSFontInfoXML (FileName As String) As String**
  - used to download a DBCS font to the POS Printer
  - returns an XML string containing an encoded DBCSFontInfo object, or an empty string if it fails

DirectIOFontInfo

- **Function GetDirectIOFontInfo(XML As String, Station As Long, FontNumber As Long, FileName As String) As String**
  - extracts the Station, FontNumber and FileName fields from a DirectIOFontInfo object encoded in an XML string
  - returns the string `Success`, or a string with the reason if it fails
  - not needed by the application but is provided for convenience
- **Function GetDirectIOFontInfoObject(XML As String, obj As DirectIOFontInfo) As String**
  - converts an XML string containing an encoded DirectIOFontInfo object into a DirectIOFontInfo object
  - returns the string `Success`, or a string with the reason if it fails
  - not needed by the application but is provided for convenience
- **Function GetDirectIOFontInfoXML(Station As Long, FontNumber As Long, FileName As String) As String**
  - used to download a font (proportional or non-proportional) to the printer
  - returns an XML string containing an encoded DirectIOFontInfo object or empty string if it fails

FiscalInformation

- **Function GetFiscalInformation(XML As String, FiscalCountry As Long, FiscalPowerInterrupted As Boolean, FiscalVersion As Long) As String**
  - used to get the FiscalCountry, FiscalPowerInterrupted, and FiscalVersion data from the Fiscal Printer; (GetFiscalInformationObject can also be used)
  - extracts the FiscalCountry, FiscalPowerInterrupted and FiscalVersion fields from a FiscalInformation object encoded in an XML string
  - returns the string `Success`, or a string with the reason if it fails
- **Function GetFiscalInformationObject(XML As String, obj As FiscalInformation) As String**
  - used to get the FiscalCountry, FiscalPowerInterrupted, and FiscalVersion data from the Fiscal Printer using the FiscalInformation object; (GetFiscalInformation can also be used)
  - converts an XML string containing an encoded FiscalInformation object into a FiscalInformation object
  - returns the string `Success`, or a string with the reason if it fails
- **Function GetFiscalInformationXML(FiscalCountry As Long, FiscalPowerInterrupted As Boolean, FiscalVersion As Long) As String**
  - returns an XML string containing an encoded FiscalInformation object or empty string if it fails
  - not needed by the application but is provided for convenience
- **Function GetByteArray(XML As String, byteArray() As Byte) As String**
  - used to do Fiscal_Read and Fiscal_Write DirectIO calls
  - converts an XML string containing an encoded byte array into an array of bytes
  - returns the string `Success`, or a string with the reason if it fails
- **Function GetByteArrayXML(byteArray() As Byte) As String**
  - used to do Fiscal_Read and Fiscal_Write DirectIO calls
  - converts an array of bytes into an XML string that contains the array of bytes encoded

Note: The byte array passed must have the correct number of elements for a Fiscal_Read operation. The proper number of elements is passed as an argument to the event handler when a Fiscal_Data_Avail directIO event is fired. Fiscal_Write is sent to the fiscal printer so the length is based on the fiscal command.

PrintGetDirectIOFontInfo

- **Function GetPrintGetDirectIOFontInfo(XML As String, receiptCharacterSet() As Long, slipCharacterSet() As Long, dBCSCodePage As Byte, matrixImpactCodePage As Byte, matrixUDThermalCodePage1() As Byte, matrixUDThermalCodePage2() As Byte, matrixUDThermalCodePage3() As Byte, matrixUDThermalCodePage4() As Byte, matrixUDImpactCodePage1() As Byte, matrixUDImpactCodePage2() As Byte, dBCSAddressVector1 As Long, dBCSAddressVector2 As Long, dBCSAddressVector3 As Long) As String**
  - used to get the Font Info data from the Fiscal Printer, (GetPrintGetDirectIOFontInfoObject can also be used)
  - extracts the all of the fields from a PrintGetDirectIOFontInfo object encoded in an XML string

  - returns `Success` when it works or a string with the reason if it fails
- **Function GetPrintGetDirectIOFontInfoObject(XML As String, obj As PrintGetDirectIOFontInfo) As String**
  - used to get the Font Info data from the Fiscal Printer, (GetPrintGetDirectIOFontInfoObject can also be used)
  - converts an XML string containing an encoded PrintGetDirectIOFontInfo object into a PrintGetDirectIOFontInfo object
  - returns `Success` when it works or a string with the reason if it fails
- **Function GetPrintGetDirectIOFontInfoXML(receiptCharacterSet() As Long, slipCharacterSet() As Long, dBCSCodePage As Byte, matrixImpactCodePage As Byte, matrixUDThermalCodePage1() As Byte, matrixUDThermalCodePage2() As Byte, matrixUDThermalCodePage3() As Byte, matrixUDThermalCodePage4() As Byte, matrixUDImpactCodePage1() As Byte, matrixUDImpactCodePage2() As Byte, dBCSAddressVector1 As Long, dBCSAddressVector2 As Long, dBCSAddressVector3 As Long) As String**
  - returns an XML string containing an encoded PrintGetDirectIOFontInfo object or empty string if it fails
  - not needed by the application but is provided for convenience

Chapter 6. Cash drawer

Cash drawer supported devices

Table 26. Cash drawer supported devices

Device	Connectivity
1. SurePOS 500/600 family built-in cash drawer	EIA-232
2. SurePOS 700 family built-in cash drawer	Integrated
3. 4694 family built-in cash drawer	RS-485
4. 4610 printer built-in cash drawer	EIA-232, RS-485, USB OEM
5. SurePOS 300 family	Integrated
6. SureOne family built-in cash drawer (OPOS)	Integrated

Supported properties and methods

Table 27. Cash drawer common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support STANDARD except SurePOS 500/600 family built-in cash drawer, which supports ADVANCED
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	Not supported
Claimed	All	
DataCount	Not supported	
DataEventEnabled	Not supported	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
PowerNotify	All	

Table 27. Cash drawer common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
PowerState	All	
State	All	

Table 28. Cash drawer specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapStatus	All	
CapStatusMultiDrawerDetect	All except 4610 printer built-in cash drawer	
DrawerOpened	All	

Table 29. Cash drawer common methods

Method	JavaPOS and OPOS Gateway	OPOS
compareFirmwareVersion	Not supported	
checkHealth	All	All except SurePOS 500/600 family built-in cash drawer
claim	All	
clearInput	Not supported	
clearOutput	All	Not supported
close	All	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 30. Cash drawer specific methods

Method	JavaPOS and OPOS Gateway	OPOS
openDrawer	All	
waitForDrawerClose	All	

Table 31. Cash drawer events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
StatusUpdateEvent	All	

JavaPOS configuration

signalsReversed

```
<Prop name="signalsReversed" type="Boolean" value="false"/>
```

Reverses the wiring polarity of the cash drawer when the cash drawer is attached to a 4610 printer. Some OEM drawers are wired in reverse to the standard IBM drawer, so will require this value to be changed to TRUE.

Valid values are TRUE and FALSE. Default is FALSE.

deviceNumber

```
<prop name="com.ibm.posj.bus.deviceNumber" type="String" value="0"/>
```

Used to properly identify the cash drawers present. The values 0 - 3 are reserved for the cash drawer attached to the system unit. For example, 0 means the cash drawer attached to port 3A on the system, and 1 means the cash drawer attached to port 3B on the system. Cash drawers attached to RS-485 and USB 4610 printers will have a deviceNumber starting with 4.

Additional JavaPOS information

4610 printer-attached cash drawer (EIA-232, RS-485 and USB)

JavaPOS supports regular IBM cash drawers and OEM cash drawers attached to the IBM 4610 Printer.

For an IBM cash drawer, set the signalsReversed property to *false*. For example:

```
<Prop name="signalsReversed" type="Boolean" value="false"/>
```

For some OEM cash drawers, the signalsReversed property will need to be set to *true*. For example:

```
<Prop name="signalsReversed" type="Boolean" value="true"/>
```

Note: In the jpos.xml.<os>Sample, the `signalsReversed` property is in the JposEntries with the following logicalName values:

- ″CashDrawer RS232 4610-A″
- ″CashDrawer RS232 4610-B″
- ″CashDrawer RS485 4610-A″
- ″CashDrawer RS485 4610-B″
- ″CashDarwer USB 4610-A″
- ″CashDrawer USB 4610-B″

Loading the Cash Drawer device driver for Linux

To load the correct Cash Drawer driver in Linux, the `/opt/ibm/javapos/etc/machine.conf` file needs to be modified. The format of `machine.conf` is:

```
<keyword> <number of entries>
<machine type><model number>
```

″<keyword>″ represents the device for which a driver is needed and ″<number of entries>″ indicates the number of ″<machine type><model number>″ pairs that

follow. If the system on which JavaPOS is running matches one of the specified ″<machine type><model number>″ pairs, the device driver for that ″<machine type><model number>″ will be loaded.

For the cash drawer, `machine.conf` might have the following entries:

```
CD 1
4810321
```

OPOS configuration

Table 32. Service Object settings for cash drawer

Keyword	Type	Description	OPOS Gateway Support
CashDrawerNumber	String	Cash drawer number. Valid values are 1 (default) and 2.	No
PulseWidthOnTime	String	Pulse on time (0-512) for firing the cash drawer. The default value is 100. **Note:** This setting cannot be modified with the Configuration tool.	No
PulseWidthOffTime	String	Pulse off time (0-512) for firing the cash drawer. The default value is 100. **Note:** This setting cannot be modified with the Configuration tool.	No
SignalsReversed	String	Reverses the wiring polarity of the cash drawer. Valid values are: **True** Reverse polarity **(other value)** Standard polarity (default)	No

Additional OPOS information

OEM Cash drawer support

Some OEM cash drawers are wired the opposite of IBM cash drawers. Use the Signals Reversed check box in the configuration utility to indicate that your cash drawer is one of these OEM cash drawers. OPOS support for the USB SureMark printer assumes the IBM cash drawer wiring. The OPOS support for EIA-232 SureMark printers prior to release 1.7.1 assumes the OEM wiring. Starting with release 1.7.1 OPOS support for the EIA-232 SureMark Printer cash drawer has been changed to match the behavior of the USB Printer cash drawer. Therefore, it may be necessary to reconfigure your EIA-232 SureMark printer cash drawer modifying the Signals Reversed check box.

Chapter 7. Check scanner

Check scanner supported devices

Table 33. Check scanner supported devices

Device	Connectivity
1. 4610 TI8 CheckScanner	EIA-232, RS-485, USB
2. 4610 TI9 CheckScanner	EIA-232, RS-485, USB

Supported properties and methods

Table 34. Check scanner common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	All	Supported
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	Supported
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	Supported
Claimed	All	Supported
CompareFirmwareVersion	Not supported	
DataCount	All	Supported
DataEventEnabled	All	Supported
DeviceControlDescription	All	Supported
DeviceControlVersion	All	Supported
DeviceEnabled	All	Supported
DeviceServiceDescription	All	Supported
DeviceServiceVersion	All	Supported
FreezeEvents	All	Supported
OutputID	All	Supported
PowerNotify	All	Supported
PowerState	All	Supported
PhysicalDeviceDescription	All	Supported
PhysicalDeviceName	All	Supported
State	All	Supported

Table 35. Check scanner specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapAutoContrast	False	
CapContrast	True	
CapAutoGenerateFileID	False	Not supported
CapAutoGenerateImageTagData	False	Not supported
CapAutoSize	True	Supported
CapColor	CheckScannerConst.CHK_CCL_GRAYSCALE	Supported
CapConcurrentMICR	True	Supported
CapDefineCropArea	True	Supported
CapImageFormat	• CheckScannerConst.CHK_CIF_TIFF • CheckScannerConst.CHK_CIF_BMP • CheckScannerConst.CHK_CIF_JPEG • CheckScannerConst.CHK_CIP_NATIVE	Supported: • CHK_CIF_TIFF • CHK_CIF_BMP • CHK_CIF_JPEG • CHK_CIP_NATIVE
CapImageTagFormat	True	Supported
CapMICRDevice	True	Supported
CapStoreImageFiles	True	Supported
CapValidationDevice	False	Not supported
Color	CheckScannerConst.CHK_CL_GRAYSCALE	CHK_CCL_GRAYSCALE only
ConcurrentMICR	True	Supported
Contrast	0 - 100, default = 50	50
CropAreaCount	0	Supported
DocumentHeight	Set by the driver after document scanned Initialized value: 8000	Supported
DocumentWidth	Set by the driver after document scanned Initialized value: 4000	Supported
FileID	""	Not supported
FileIndex	0	Supported
ImageData	NULL	Supported
ImageFormat	Supported: CheckScannerConst.CHK_CIF_NATIVE CheckScannerConst.CHK_CIF_TIFF CheckScannerConst.CHK_CIF_BMP CheckScannerConst.CHK_CIF_JPEG	Supported: • CHK_IF_TIFF • CHK_IF_JPEG • CHK_IF_BMP • CHK_IF_NATIVE
ImageMemoryStatus	Current state of checkscanner memory	Supported
ImageTagData	Max 32 characters allowed	Supported
MapMode	Only CheckScannerConst.CHK_MM_ENGLISH supported	Supported: CHK_MM_ENGLISH only
MaxCropAreas	20	Supported
Quality	Only 200 dpi supported	TI8: 200, TI9: 200 (default) or 100
QualityList	1 - set to ″100″ 2 - set to "100 200"	TI8: "200", TI9: "100,200"

Table 35. Check scanner specific properties (continued)

Property	JavaPOS and OPOS Gateway			OPOS	
RemainingImagesEstimate *(Estimates when memory is empty.)*		**TI8/TI9**	**TI9**	**TI8/TI9**	**TI/9**
	Format:	200 DPI	100 DPI	200 DPI	100 DPI
	CheckScannerConst. CHK_CIF_TIFF	108	216	108	216
	CheckScannerConst. CHK_CIF_BMP	1	2	1	2
	CheckScannerConst. CHK_CIF_JPEG	20	40	20	40
	CheckScannerConst. CHK_CIF_NATIVE	1	2	1	2

Table 36. Check scanner common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	Supported: JposConst.JPOS_CH_INTERNAL JposConst.JPOS_CH_EXTERNAL Throws jposException(JPOS_E_ILEGAL) JposConst.JPOS_CH_INTERACTIVE	All Supported: OPOS_CH_INTERNAL OPOS_CH_EXTERNAL OPOS_CH_INTERACTIVE
claim	All	Supported
clearInput	All	Supported
clearInputProperties	All	Supported
clearOutput	All	Supported
close	All	Supported
directIO	All	Supported
open	All	Supported
release	All	Supported
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 37. Check scanner specific methods

Method	JavaPOS and OPOS Gateway	OPOS
beginInsertion	All	Supported
beginRemoval	All	Supported
clearImage	All (Only CheckScannerConst.CHK_CLR_ALL supported.)	Supported: CHK_CLR_ALL only
defineCropArea	All	Supported
endInsertion	All	Supported
endRemoval	All	Supported

Table 37. Check scanner specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
retrieveImage	Supported: CheckScannerConst.CHK_CROP_ AREA_ENTIRE_IMAGE. All other values are supported starting from EC level 0x68	Supported: CHK_CROP _AREA_ENTIRE _IMAGE All other values are supported starting from EC level 0x68
retrieveMemory	Supported: CheckScannerConst.CHK_LOCATE_ BY_FILEINDEX. All other values throw JposException(JPOS_E_ILLEGAL)	Supported: CHK_LOCATE_BY_FILEINDEX CHK_LOCATE_BY_IMAGETAGDATA. Not suppported: CHK_LOCATE_BY_FILEID returns OPOS_E_ILLEGAL.
storeImage	All	

Table 38. Check scanner events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	Supported
DirectIOEvent	All	Supported
ErrorEvent	All	Supported
StatusUpdateEvent	All	Supported

JavaPOS DirectIO calls

The following directIO commands are supported for the check scanner device. The syntax is as follows:

```
directIO ( command: int32, inout data: int32 , inout obj: object):
void { raises-exception }
```

To access DirectIO constants: import `com.ibm.jpos.services.DirectIO`.

Scanner calibration command

Calibrate the check scanner in IBM 4610 Printer Models TI8 and TI9.

Table 39. CHK_DIO_SCANNER_CALIBRATION_CMD information

Parameter	Type	Value
Command	Int32	DirectIO.CHK_DIO_SCANNER_CALIBRATION_CMD
Data	Int32	Any value. A null value is accepted.
Obj	Object	Any value. A null value is accepted.

Remarks

A white document must be inserted in the DI station. The printer scans the document two or three times to update the printer's calibration data. Then directIO is paired with the beginInsertion method for controlling check insertion.

Print scanned image command

Prints the specified scanned image to the thermal station.

Table 40. CHK_DIO_PRINT_SCANNED_IMAGE_CMD information

Parameter	Type	Value
Command	Int32	DirectIO.CHK_DIO_PRINT_SCANNED_IMAGE_CMD
Data[0]	Int32	DirectIO.CHK_DIO_MEMORY_IMG prints the most recently scanned image. DirectIO.CHK_LOCATE_BY_FILEINDEX prints the image file using the FileIndex property.
Data[1]	Int32	The numeric identifier for the defined crop area. If the values is CHK_CROP_AREA_ENTIRE__IMAGE then the entire area of the scanned image is printed.
Data[2]	Int32	Percentage to scale image in x direction. (A value of 100 or 0 results in no scaling.)
Data[3]	Int32	Percentage to scale image in y direction. (A value of 100 or 0 results in no scaling.)
Data[4]	Int32	DirectIO.CHK_DIO_ROTATE_90 rotates the image counter clockwise 90%. DirectIO.CHK_DIO_NO_ROTATE does not rotate the image.
Obj	Object	Any value. A null value is accepted.

Remarks

Printing stored images causes the printer to pause momentarily while the printer formats the data to be printed. This time varies depending on the amount of formatting required.

If cx and cy defined at the crop area extend the printer area further than the boundaries of the image, the values are truncated to the image boundary. If the size of the print area is greater than the thermal print head is capable of printing (either from image size or scaling), the image is truncated to the width of the print head. Currently, the only scaling options are 100% (no scaling) and 200% (double). Printing resident images in the work area memory with CheckScannerConst.CHK_IF_JPEG format is not supported.

Set check side command

Set the side of the check to scan. This command assumes that the check is inserted face down.

Table 41. CHK_DIO_SET_CHECK_SIDE_CMD information

Parameter	Type	Value
Command	Int32	DirectIO.CHK_DIO_SET_CHECK_SIDE_CMD
Data[0]	Int32	• DirectIO.CHK_DIO_SIDE1 selects the front of the check. • DirectIO.CHK_DIO_SIDE2 selects the back of the check. • DirectIO.CHK_DIO_OPPOSITE selects the opposite face of the check from the current side selected.
Obj	Object	Any value. A null value is accepted.

Remarks

This allows a check scan to occur on both sides of the document. If a document is not inserted, an error is returned. If data[0] is different from the value returned by DirectIO.CHK_DIO_GET_CHECK_SIDE_CMD or data[0] is DirectIO.CHK_DIO_SIDE_OPPOSITE, the side of the document is

changed and the document is fed. If data[0] is the same as the value returned by DirectIO.CHK_DIO_GET_CHECK_SIDE_CMD, nothing occurs and the method returns.

Get check side command

Return the current setting for the side of the check to be scanned. This command assumes the check is inserted face down. After calling this DirectIO, data[0] is populated with one of the values in the following table:

Table 42. CHK_DIO_GET_CHECK_SIDE_CMD information

Parameter	Type	Value
Command	Int32	DirectIO.CHK_DIO_GET_CHECK_SIDE_CMD
Data	Int32	• DirectIO.CHK_DIO_SIDE_UNKNOWN (Indicates no check is inserted.) • DirectIO.CHK_DIO_SIDE1 (Indicates the front of the check is selected.) • DirectIO.CHK_DIO_SIDE2 (Indicates the back of the check is selected.)
Obj	Object	Any value. A null value is accepted.

Remarks

This property value can be changed using the DirectIO.CHK_DIO_SET_CHECK_SIDE_CMD method. When a check is inserted the value returned is DirectIO.CHK_DIO_SIDE1.

Scanner image quality command

Check the quality of the image scanned on Model TI9 printers. The check scanner automatically checks the image quality of each check scanned against a set of internal quality attributes. After calling this DirectIO, data[0] is populated with a value of 0 or 1.

- 0 indicates image quality is acceptable.
- 1 indicates image quality may not be acceptable.

Table 43. CHK_DIO_SCANNER_IMAGE_QUALITY_CMD information

Parameter	Type	Value
Command	Int32	DirectIO.CHK_DIO_SCANNER_IMAGE_QUALITY_CMD
Data	Int32	An array with at least one element
Obj	Object	Any value. A null value is accepted.

Remarks

This directIO is supported only for the 4610 TI9 model.

Additional OPOS information

Multiple devices

A POS machine can not have both RS-485 and USB checkscanners operating at the same time.

Chapter 8. Fiscal printer

Fiscal printer operations are supported through directIO() calls. The directIO() functions do not perform any verification of commands sent to the fiscal printer or of data returned from the printer. The functions provide a simple pass-through operation, allowing an application to access the fiscal functions on the printer.

Fiscal printer supported devices

USB Fiscal printers are not supported on Linux.

Table 44. Fiscal printer supported devices

IBM SureMark Printer Fiscal Printer Device	Connectivity	Comments
1. Models Kx3	EIA-232	KD3: Hungary, Italy, Czech Republic, Romania
2. Models Kx4	EIA-232	KJ4: Brazil KN4: Brazil KC4: Chile
3. Models Kx5	EIA-232	KC5: Chile KD5: Hungary, Italy, Czech Republic, Romania
4. Models Kx3	RS-485	KR3: Hungary, Italy, Czech Republic, Greece, Turkey
5. Models Kx5	RS-485	KR5: Hungary, Italy, Czech republic, Greece, turkey KR5: Chile
6. Models Kx5	USB (no Linux support)	KB5: Turkey KH5: Greece
7. Models G*x*3	EIA-232	
8. Models Gx3	RS-485	GR3: Argentina, Greece, Italy, Turkey
9. Models Gx3	USB	GB3: Greece, Turkey GE3: Argentina
10. Models Gx4	EIA-232	GD4: Venezuala
11. Models Gx5	EIA-232	GD5: Turkey
12. Models Gx5	RS-485	GF5: Greece, Italy, Turkey
13. Models Gx5	USB	GB5: Greece, Turkey
14. Models Sx6	EIA-232	SJ6: Brazil
15. Models Sx6	USB	SJ6: Brazil
16. Models 3xA	RS-485	3FA: Argentina
17. Models Kx4	RS-485	KS4: Chile
18. Models K*x*3	USB	KB3: Turkey
19. G*x*4	RS-485	GR4: Venezuala
20. 3B*x*	RS-485	3BS: Brazil 3BM: Brazil
21. 3F	RS-485	Brazil, Bulgaria, Greece, Hungary, Italy, Latvia, Mexico, Romania, Russia, Slovakia, turkey, Uruguay, Venezuala
Country-independent models (GF3/GF5, GG3/GG5, KF3/KF5, KG3/KG5) are not listed.		

Supported properties and methods

Table 45. Fiscal printer common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	Not supported
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	Not supported
Claimed	All	Not supported
DataCount	Not supported	
DataEventEnabled	Not supported	
DeviceControlDescription	All	Not supported
DeviceControlVersion	All	Not supported
DeviceServiceDescription	All	Not supported
DeviceServiceVersion	All	Not supported
FreezeEvents	All	Not supported
OutputID	All	Not supported
PhysicalDeviceDescription	All	Not supported
PhysicalDeviceName	All	Not supported
PowerNotify	PN_DISABLED	Not supported
PowerState	PS_UNKNOWN	Not supported
State	All	Not supported

Table 46. Fiscal printer specific properties

Property	JavaPOS	OPOS
ActualCurrency	E_ILLEGAL	Not supported
AdditionalHeader	E_ILLEGAL	Not supported
AdditionalTrailer	E_ILLEGAL	Not supported
AmountDecimalPlaces	E_ILLEGAL	Not supported
AsyncMode	All - false	Not supported
CapAdditionalHeader	All - false	Not supported
CapAdditionalLines	All - false	Not supported
CapAdditionalTrailer	All - false	Not supported
CapAmountAdjustment	All - false	Not supported
CapAmountNotPaid	All - false	Not supported
CapChangeDue	All - false	Not supported
CapCheckTotal	All - false	Not supported

Table 46. Fiscal printer specific properties (continued)

Property	JavaPOS	OPOS
CapCoverSensor	All - false	Not supported
CapDoubleWidth	All - false	Not supported
CapDuplicateReceipt	All - false	Not supported
CapEmptyReceiptIsVoidable	All - false	Not supported
CapFiscalReceiptStation	All - false	Not supported
CapFiscalReceiptType	All - false	Not supported
CapFixedOutput	All - false	Not supported
CapHasVatTable	All - false	Not supported
CapIndependentHeader	All - false	Not supported
CapItemList	All - false	Not supported
CapJrnEmptySensor	All - false	Not supported
CapJrnNearEndSensor	All - false	Not supported
CapJrnPresent	All - false	Not supported
CapMultiContractor	All - false	Not supported
CapNonFiscalMode	All - false	Not supported
CapOnlyVoidLastItem	All - false	Not supported
CapOrderAdjustmentFirst	All - false	Not supported
CapPackageAdjustment	All - false	Not supported
CapPercentAdjustment	All - false	Not supported
CapPositiveAdjustment	All - false	Not supported
CapPostPreLine	All - false	Not supported
CapPowerLossReport	All - false	Not supported
CapPredefinedPaymentLines	All - false	Not supported
CapReceiptNotPaid	All - false	Not supported
CapRecEmptySensor	All - true	Not supported
CapRecNearEndSensor	All - false	Not supported
CapRecPresent	All - true	Not supported
CapRemainingFiscalMemory	All - false	Not supported
CapReservedWord	All - false	Not supported
CapSetCurrency	All - false	Not supported
CapSetHeader	All - false	Not supported
CapSetPOSID	All - false	Not supported
CapSetStoreFiscalID	All - false	Not supported
CapSetTrailer	All - false	Not supported
CapSetVatTable	All - false	Not supported
CapSlpEmptySensor	All - false	Not supported
CapSlpFiscalDocument	All - false	Not supported
CapSlpFullSlip	All - false	Not supported
CapSlpNearEndSensor	1 and 2 - false 3 - true	Not supported

Table 46. Fiscal printer specific properties (continued)

Property	JavaPOS	OPOS
CapSlpPresent	1 and 2 - false 3 - true	Not supported
CapSlpValidation	All - false	Not supported
CapSubAmountAdjustment	All - false	Not supported
CapSubPercentAdjustment	All - false	Not supported
CapSubtotal	All - false	Not supported
CapTotalizerType	All - false	Not supported
CapTrainingMode	All - false	Not supported
CapValidateJournal	All - false	Not supported
CapXReport	All - false	Not supported
ChangeDue	E_ILLEGAL	Not supported
CheckTotal	E_ILLEGAL	Not supported
ContractorId	E_ILLEGAL	Not supported
CountryCode	E_ILLEGAL	Not supported
CoverOpen	All - true or false	Not supported
DateType	E_ILLEGAL	Not supported
DayOpened	E_ILLEGAL	Not supported
DescriptionLength	E_ILLEGAL	Not supported
DuplicateReceipt	E_ILLEGAL	Not supported
ErrorLevel	E_ILLEGAL	Not supported
ErrorOutID	E_ILLEGAL	Not supported
ErrorState	E_ILLEGAL	Not supported
ErrorStation	E_ILLEGAL	Not supported
ErrorString	E_ILLEGAL	Not supported
FiscalReceiptStation	E_ILLEGAL	Not supported
FiscalReceiptType	E_ILLEGAL	Not supported
FlagWhenIdle	E_ILLEGAL	Not supported
JrnEmpty	E_ILLEGAL	Not supported
JrnNearEnd	E_ILLEGAL	Not supported
MessageLength	E_ILLEGAL	Not supported
MessageType	E_ILLEGAL	Not supported
NumHeaderLines	E_ILLEGAL	Not supported
NumTrailerLines	E_ILLEGAL	Not supported
NumVatRates	E_ILLEGAL	Not supported
PostLine	E_ILLEGAL	Not supported
PredefinedPaymentLines	E_ILLEGAL	Not supported
PreLine	E_ILLEGAL	Not supported
PrinterState	E_ILLEGAL	Not supported
QuantityDecimalPlaces	E_ILLEGAL	Not supported
QuantityLength	E_ILLEGAL	Not supported

Table 46. Fiscal printer specific properties (continued)

Property	JavaPOS	OPOS
RecEmpty	All - true or false	Not supported
RecNearEnd	E_ILLEGAL	Not supported
RemainingFiscalMemory	E_ILLEGAL	Not supported
ReservedWord	E_ILLEGAL	Not supported
SlpEmpty	All - true or false	Not supported
SlpNearEnd	E_ILLEGAL	Not supported
SlipSelection	E_ILLEGAL	Not supported
TotalizerType	E_ILLEGAL	Not supported
TrainingModeActive	E_ILLEGAL	Not supported

Table 47. Fiscal printer common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	Not supported
claim	All	Not supported
clearInput	Not supported	
clearOutput	Not supported	
close	All	Not supported
compareFirmwareVersion	E_ILLEGAL	E_ILLEGAL
directIO	All	Not supported
open	All	Not supported
release	All	Not supported
resetStatistics	E_ILLEGAL	E_ILLEGAL
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	E_ILLEGAL
updateFirmware	E_ILLEGAL	E_ILLEGAL
updateStatistics	E_ILLEGAL	E_ILLEGAL

Table 48. Fiscal printer specific methods

Method	JavaPOS and OPOS Gateway	OPOS
None supported	The communication to Fiscal Printer is only through directIO.	Not supported

Table 49. Fiscal printer events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All - yes	Not supported
ErrorEvent	Not supported	
OutputCompleteEvent	Not supported	
StatusUpdateEvent	Not supported	

JavaPOS DirectIO calls

The following directIO commands are supported for the fiscal printer device. The syntax is as follows:

```
directIO ( command: int32, inout data: int32 , inout obj: object):
void { raises-exception }
```

To access FiscalPrinter constants, import: `com.ibm.jpos.services.*`

Fiscal information command

Obtain information about the fiscal device. Valid values are returned only after the first FISCAL_IPL_END_STATUS event.

Table 50. Parameters for FISCAL_INFORMATION subcommand

Parameter	Type	Value
Command	Int32	DirectIO.FISCAL_ID
Data	Int32	DirectIO.FISCAL_INFORMATION
Obj	Object	FiscalInformation Object

Remarks

For the FiscalInformation object,

- getCountry() returns the Country Code value that is specified in the Fiscal Printer Hardware Supplements.
- getVersion() returns the microcode version (EC level).
- getFiscalPowerInterrrupted() returns `true` if the printer was turned Off in the middle of a command; otherwise, it returns `false`. (For more information refer to the Fiscal Printer Hardware Supplements available on the Web at http://www.ibm.com/solutions/retail/store .)

Fiscal read command

Read data from the fiscal device. This subcommand should be issued after receiving a FISCAL_DATA_AVAIL directIOEvent from the fiscal device.

Table 51. Parameters for FISCAL_READ subcommand

Parameter	Type	Value
Command	Int32	com.ibm.jpos.services.DirectIO.FISCAL_ID
Data	Int32	com.ibm.jpos.services.DirectIO.FISCAL_READ
Obj	Object	A byte [ ] to be filled with data.

Remarks

If a FISCAL_DATA_AVAIL event is received and the application has not read the previous fiscal data, the old data is replaced by the new incoming data. The next FISCAL_READ command reads the new data.

Fiscal write command

Write data to the fiscal device.

Table 52. Parameters for FISCAL_WRITE subcommand

Parameter	Type	Value
Command	Int32	DirectIO.FISCAL_ID
Data	Int32	DirectIO.FISCAL_WRITE
Obj	Object	A byte [ ] with the data to send. The byte [ ] (object parameter) must contain the exact data to send. This data is a Fiscal Command without the prefix {X'1B',X'66'}.

Remarks

DirectIO is a synchronous method and returns successfully when the data is written to the device; however, command execution is not complete until a FISCAL_STATUS or FISCAL_ERROR event is received. If another command is submitted before the reception of these events, the second command is ignored. This means that there is no buffering of FISCAL_WRITE subcommands.

Fiscal notify command

Set fiscal notification On or Off. If fiscal notification is On, every fiscal command returns a status event to the application.

Table 53. Parameters for FISCAL_NOTIFY subcommand

Parameter	Type	Value
Command	Int32	DirectIO.FISCAL_ID
Data	Int32	DirectIO.FISCAL_NOTIFY
Obj	Object	An Integer object with one of the following values: DirectIO.FISCAL_NOTIFY_ON or DirectIO.FISCAL_NOTIFY_OFF

JavaPOS DirectIO events

Fiscal error event

Indicates that a fiscal error has occurred. The DirectIOEvent parameters are shown in Table 54.

Table 54. Parameters for the FISCAL_ERROR event

Parameter	Type	Value
Event Number	Int32	DirectIO.FISCAL_ERROR
Data	Int32	Cause of the fiscal error event (Fiscal Printer Command Return Code).
Obj	Object	Integer object that indicates the error codes as in the POS Subsystem (see Table 55). The Integer value is the result of an OR operation of the current error codes (printer status).

Table 55. Error codes for FISCAL_ERROR event

Error code	Value
PosSTATUS_COVER_OPEN	X'0001'
PosSTATUS_TRANSPORT_ERROR	X'0002'
PosSTATUS_SJ_PAPER_ERROR	X'0004'
PosSTATUS_DOCUMENT_AT_FRONT	X'0008'
PosSTATUS_DOCUMENT_AT_TOP	X'0010'
PosSTATUS_DOCUMENT_READY	X'0020'
PosSTATUS_HEAD_PARKED	X'0040'
PosSTATUS_INSERTED_FORWARD	X'0080'
PosSTATUS_ERROR_PENDING	X'0100'
PosSTATUS_DI_FRONT_LOAD_ERROR	X'0200'
PosSTATUS_DI_TOP_LOAD_ERROR	X'0400'
PosSTATUS_PRINTER_ONLINE	X'1000'
PosSTATUS_MICR_INSTALLED	X'2000'
PosSTATUS_CR_PAPER_LOW	X'4000'
PosSTATUS_SJ_PAPER_LOW	X'8000'

Fiscal status event

Indicates that the last fiscal command completed successfully. Fiscal notification must be On to receive these events. The DirectIOEvent parameters are shown in Table 56.

Table 56. Parameters for the FISCAL_STATUS event

Parameter	Type	Value
Event Number	Int32	DirectIO.FISCAL_STATUS
Data	Int32	Always 0
Obj	Object	Integer object that indicates the last command executed. For example for the command {X'FF', X'10', '0', '0', '0', '0', '0', '0', '0', '0', '0', 'F'}, the object returns the value X'FF' (the first byte indicating the command.) See the Fiscal Printer Hardware Supplements available on the Web at http://www.ibm.com/solutions/retail/store for more information.

Fiscal data available event

Indicates that data is available to be read from the fiscal device. The DirectIOEvent parameters are shown in Table 57.

Table 57. Parameters for the FISCAL_DATA_AVAIL event

Parameter	Type	Value
Command	Int32	DirectIO.FISCAL_DATA_AVAIL
Data	Int32	Always 0
Obj	Object	Integer object that indicates the size in bytes of the data to be read.

Fiscal raw status event

Indicates status has been received. The directIO parameters are shown in Table 58.

Table 58. Parameters for the FISCAL_RAW_STATUS event

Parameter	Type	Value
Command	Int32	DirectIO.FISCAL_RAW_STATUS
Data	Int32	Always 0
Obj	Object	Byte [ ] object containing the raw status data

Remarks

The object parameter is filled with bytes reported by the Fiscal Printer Device. The processing of these bytes must be done according to the Fiscal Printer Hardware Supplements available on the Web at http://www.ibm.com/solutions/retail/store.

Fiscal IPL end status event

Indicates that the IPL process is complete and that the application can start sending commands to the Fiscal Printer.

The directIO parameters are shown in Table 59.

Table 59. Parameters for the FISCAL_IPL_END_STATUS event

Parameter	Type	Value
Event Number	Int32	DirectIO.FISCAL_IPL_END_STATUS
Data	Int32	Always 0
Obj	Object	Always null

Remarks

When the Fiscal Printer is reset or powered on, an initial program load (IPL) occurs. During IPL, the printer is not operational and rejects fiscal commands. The IPL process can take more than one minute in some situations.

If the application tries to write a command before it receives a FISCAL_IPL_END_STATUS, the driver ignores the command.

Chapter 9. Hard totals

Hard totals supported devices

Table 60. Hard totals supported devices

Device	Connectivity
1. 4694 NVRAM	Integrated
2. SurePOS 700 Series NVRAM	Integrated
3. SureOne built-in NVRAM (excluding A04/A05 models)	Integrated

Supported properties and methods

Table 61. Hard totals common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_NONE	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	Not supported	
DeviceEventEnabled	Not supported	All
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	All	
OutputID	Not supported	All
PowerNotify	PN_DISABLED	All
PowerState	PS_UNKNOWN	All
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	All	
ResultCodeExtended	All	

Table 61. Hard totals common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
State	All	

Table 62. Hard totals specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapErrorDetection	Not supported	
CapSingleFile	All	
CapTransactions	Not supported	
FreeData	All	
NumberOfFiles	Maximum of 1 file	
TotalsSize	All	
TransactionInProgress	Always false	

Table 63. Hard totals common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	Not supported	
clearOutput	Not supported	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 64. Hard totals specific methods

Method	JavaPOS and OPOS Gateway	OPOS
beginTrans	Not supported	
claimFile	All	
commitTrans	Not supported	
create	All	
delete	All	
find	All	
findByIndex	All	
releaseFile	All	
read	All	

Table 64. Hard totals specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
recalculateValidationData	Not supported	
rename	Not supported	
rollback	Not supported	
setAll	All	
validateData	Not supported	
write	All	

Table 65. Hard totals events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
StatusUpdateEvent	All	

OPOS configuration

Note: These settings cannot be modified with the Configuration tool.

Table 66. Service object settings for hard totals

Keyword	Type	Description	OPOS Gateway Support
UseHarddiskFile	String	Enables use of file on hard driver for Hard Totals. Valid values are: **True** Use hard disk **False** Do not use hard disk (default)	No
HarddiskFileName	String	Path name of file to use for Hard Totals storage on hard disk.	No
HarddiskFileSize	String	Size (in bytes) of file used for Hard Totals Storage on hard disk.	No

Chapter 10. Keylock

Supported devices

Table 67. Keylock supported devices

Device		Connectivity
Two-position	1. Retail alphanumeric POS keyboard with card reader	PS/2, RS-485, USB
Two-position	2. Retail alphanumeric POS keyboard with MSR and pointing device	PS/2
Two-position	3. Retail POS keyboard	RS-485, USB
Two-position	4. Retail POS keyboard with card reader	RS-485, USB
Two-position	5. Retail POS keyboard with card reader and display	RS-485, USB
Two-position	6. Modifiable layout keyboard with card reader	RS-485, USB
Two-position	7. SurePoint 4820 integrated keylock	RS-485, USB
Three-position	8. SureOne integrated keyboard	PS/2
Four-position	9. POS keyboard V	RS-485, USB
Four-position	10. PC POS keyboard or PC Point of Sale keyboard	PS/2, RS-485, USB
Four-position	11. Retail POS keyboard VI	RS-485, USB
Three-position	12. SurePOS 100 integrated keyboard	PS/2
Five-position	13. 4674 Integrated keyboard	RS-485
Four-position	14. 4685-KC1 (OPOS only)	RS-485
Four-position	15. 4685-K01 (OPOS only)	RS-485
Four-position	16. 4685-K02 (Ultra7) keyboard with card reader	RS-485, USB
Four-position	17. 4685-K02 with MSR/Encoder and 4 position keylock	USB
Six-position	18. 4685-K02 with MSR/Encoder and 6 position keylock	USB
Four-position	19. 4685-K03	RS-485,USB
Two-position/ Four-position	20. Modular alphanumeric keyboard and two-position keylock	PS/2, USB
Two-position/ Four-position	21. Modular Compact alphanumeric keyboard and two-position keylock	PS/2, USB
Two-position/ Four-position	22. Modular 67 Key keyboard and two-position keylock	PS/2, USB

The JavaPOS driver will return pre-determined UPOS values for each keylock position. The OPOS driver generally allows the user to configure each key position to a different meaning.

For a 2-position keylock, the OPOS driver can either return a pre-determined UPOS value, or the raw hardware value for each keylock position (which the application can then interpret as it needs to).

For a 3-position keylock, the OPOS driver will return pre-determined UPOS values for each keylock position.

For 4-, 5-, and 6-position keylocks, the OPOS driver allows you to map the raw hardware value to any UPOS value, so that the driver can associate a physical key position with whatever UPOS value you require.

For keylocks that contain more than the 3 positions than the UPOS standard provides, the drivers will report those higher positions as a unique value. Table 68 describes the 3 standard positions in the UPOS standard.

Table 68. UPOS keylock standards

Position	Value returned
LOCK_KP_LOCK	1
LOCK_KP_NORM	2
LOCK_KP_SUPR	3

Two-position keylock

2 position (devices 1, 2, 3, 4, 5, 6, 20, 21, 22)

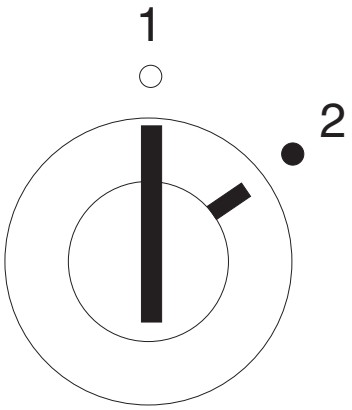


Figure 9. NANPOS, 50-key POS keyboard and 133-key POS keyboard two-position keylock

Position	JPOS value	OPOS mapped value	OPOS raw value
1	LOCK_KP_NORM	LOCK_KP_NORM	2
2	LOCK_KP_SUPR	LOCK_KP_SUPR	1

2 position (device 7)

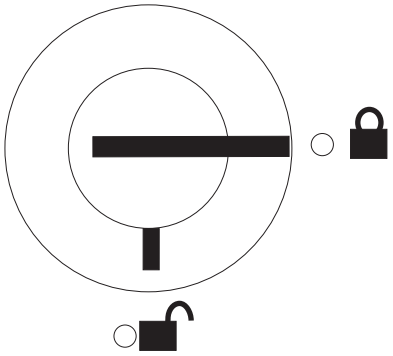


Figure 10. 4820 Two-position keylock

Position	JPOS value	OPOS mapped value	OPOS raw value
Lock	LOCK_KP_NORM	LOCK_KP_NORM	2
Unlock	LOCK_KP_SUPR	LOCK_KP_SUPR	1

Three-position keylock (devices 8, 12)

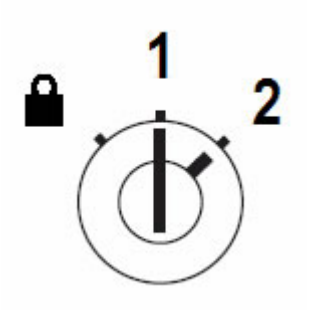


Figure 11. SureOne three-position keylock

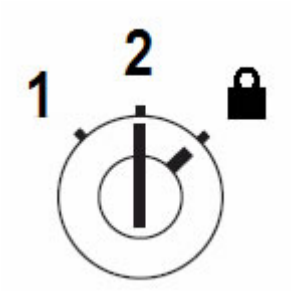


Figure 12. SurePOS 100 three-position keylock

Position	JPOS value	OPOS mapped value
LOCK	LOCK_KP_LOCK	LOCK_KP_LOCK
1	LOCK_KP_NORM	LOCK_KP_NORM
2	LOCK_KP_SUPR	LOCK_KP_SUPR

Four-position (devices 9, 10, 11, 14, 15, 16, 17, 19)

The keylocks for the ANKPOS, KeyboardV, Keyboard K02 Ultra VII (4685), the 50-Key with Jucc MSR keyboards, and Keyboard 4685-K03 all have four positions.

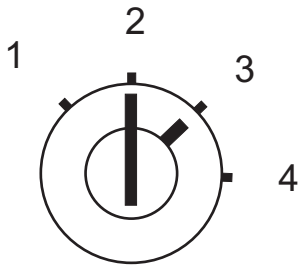


Figure 13. Four-position keylock

Position	JPOS value	OPOS mapped value	OPOS raw value
1	4	Configured by user	4
2	LOCK_KP_LOCK		3
3	LOCK_KP_NORM		2
4	LOCK_KP_SUPR		1

Four-position (devices 20, 21, 22)

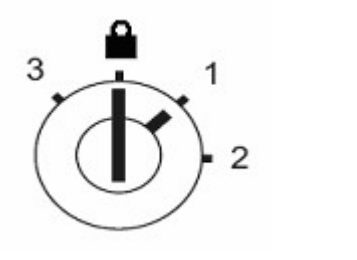


Figure 14. Four-position keylock

Position	JPOS value	OPOS mapped value	OPOS raw value
3	4	Configured by user	4
Lock	LOCK_KP_LOCK		3
1	LOCK_KP_NORM		2
2	LOCK_KP_SUPR		1

Five-position (device 13)

The keylock for the IBM 4674 keyboard has five positions.

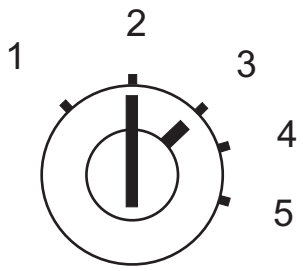


Figure 15. Five-position keylock

Position	JPOS value	OPOS mapped value	OPOS raw value
1	5	Configured by user	5
2	LOCK_KP_LOCK		3
3	LOCK_KP_NORM		2
4	LOCK_KP_SUPR		1
5	4		4

Six-position (device 18)

The keylock for the Ultra 7 keyboard with MSR Encoder has six positions. The abstraction and implementation classes are as follows:

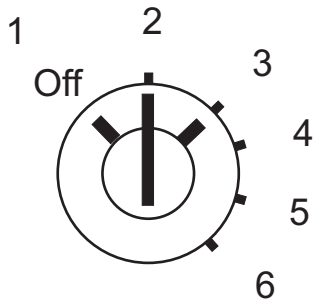


Figure 16. Six-position keylock

- com.ibm.jpos.services.IBM6PositionKeylock
- com.ibm.jpos.services.poss.IBM6PositionKeylockImp

Position	JPOS value	OPOS mapped value	OPOS raw value
1	LOCK_KP_LOCK	Configured by user	3
2	LOCK_KP_NORM		2
3	5		5
4	6		6
5	LOCK_KP_SUPR		1
6	4		4

Supported properties and methods

Table 69. Keylock common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	

Table 69. Keylock common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapPowerReporting	PR_NONE for EIA-232, PS/2 and embedded PR_STANDARD for all other keylock devices	PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	Not supported	
DataEventEnabled	Not supported	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	
OutputID	Not supported	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
State	All	

Table 70. Keylock specific properties

Property	JavaPOS and OPOS Gateway	OPOS
KeyPosition	All	
PositionCount	All	

Table 71. Keylock common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	

Table 71. Keylock common methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
claim	Always shareable	
clearInput	Not supported	
clearOutput	Not supported	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	Always shareable	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 72. Keylock specific methods

Method	JavaPOS and OPOS Gateway	OPOS
waitForKeylockChange	All	

Table 73. Keylock events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
StatusUpdateEvent	All	

JavaPOS configuration

PositionCount

```
<prop name="com.ibm.jpos.sdi.config.Keylock.PositionCount" type="String" value="4"/>
```

For some IBM keylocks, the number of keylock positions cannot be reliably determined:

- RS-485 4685-K03 - 4 positions
- RS-485 4685-K02 Keyboard with MSR Encoder - 6 positions

In these situations, use the PositionCount property to define the number of keylock positions.

Values accepted
Any value on Integer range

Device support
All

OPOS configuration

Table 74. Service Object settings for keylock

Keyword	Type	Description	OPOS Gateway Support
SecondKeyPosition	String	When present, the additional position on the Japanese ANPOS keyboard is mapped to a LOCK_KP_LOCK.	Yes
PositionCount	DWORD	4685-K02 only. Number of keylock positions (1-6).	Yes
PositionMapFrom	Binary	4685-K02 only. Keylock position code mapping (from).	Yes
PositionMapTo	Binary	4685-K02 only. Keylock position code mapping (to).	Yes
LetLastKeepAcquired	String	Allows the last interface to access the physical POS keyboard interface to keep the device acquired instead of passing it on to the next device that attempts to acquire it. **Note:** This setting is an *unsupported and untested* option used to modify Service Object behavior only in very limited cases. Use it only when directed to do so by IBM Support.	No
OnlineTimeout	String	Time in milliseconds to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.	No

Additional OPOS information

Two-position keylock

The keylocks for the NANPOS, 50–Key, 133–Key, and 4820 Point of Sale Keyboards all have two positions and report them as LOCK_KP_NORM and either LOCK_KP_SUPR or LOCK_KP_LOCK.

In OPOS, two-position keylocks can report the key positions as LOCK_KP_NORM and either LOCK_KP_SUPR or LOCK_KP_LOCK. In the OPOS Configuration Utility:

- Selecting the **IBM Raw Values** radio button during configuration will cause the manager position on the keyboard to return LOCK_KP_LOCK.
- Selecting the **Map to OPOS Values** radio button will cause the manager position to return LOCK_KP_SUPR.

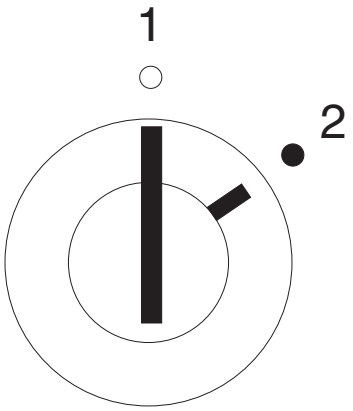


Figure 17. Two-position keylock

Key position	Description	Key position value
1	Operator	LOCK_KP_NORM
2	Manager	LOCK_KP_SUPR LOCK_KP_LOCK (OPOS only)

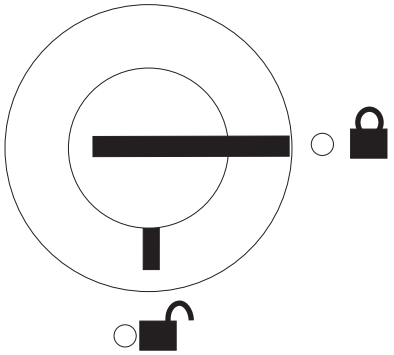


Figure 18. 4820 Two-position keylock

Key position	Description	Key position value
Lock	Operator	LOCK_KP_NORM
Unlock	Manager	LOCK_KP_SUPR LOCK_KP_LOCK (OPOS only)

Position count

For some IBM keylocks, the number of keylock positions cannot be reliably determined by OPOS:

- RS-485 4685-K03 - 4 positions
- RS-485 4685-K02 Keyboard with MSR Encoder - 6 positions

For this reason, a mechanism is provided to define the mapping of the native keylock positions to OPOS values. During configuration of the keylock device, the number of positions and the mapping of the positions can be set. Use the three-six position keyboard keylock for keylocks with 3 to 6 positions. In the OPOS Configuration Utility, use the Keylock entry labeled ″Three-Six position keyboard keylock″ or ″Three-six position keyboard keylock - USB″ for keylocks with 3 to 6

positions.

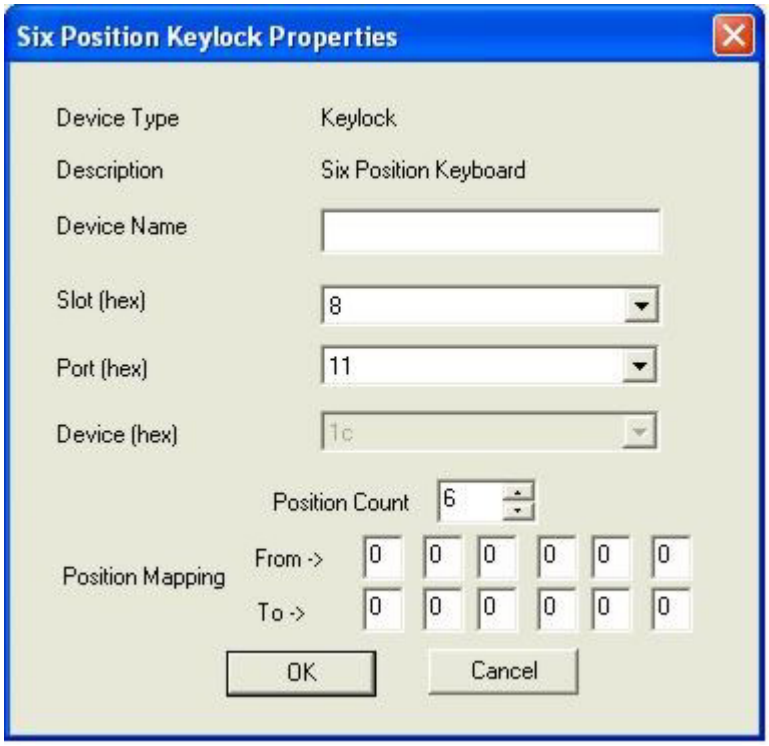


Figure 19. Three-six position keylock properties

Chapter 11. Line display

Supported devices

Table 75. Line display supported devices

Device		Connectivity
SBCS	1. 40-character VFD II/SureOne	EIA-232
	2. 40-character VFD II	USB, RS-485
	3. Two-sided VFD II	USB, RS-485
	4. 40-character VFD II (Japan)/4674	RS-485
	5. Character Graphic (C/G) APA	USB, RS-485
	6. 40-character LCD	USB, RS-485
	7. 50-key keyboard LCD	USB, RS-485
DBCS	8. Character Graphic (C/G) APA	EIA-232
	9. Character Graphic (C/G) APA	USB, RS-485
	10. PLU keyboard APA	USB, RS-485
SBCS	11. LED display 1x11	EIA-232

Note: On the SurePOS 500/600, the line display is preconfigured for COM4. Other values can be selected through the BIOS, but this is not recommended.

Supported properties and methods

Table 76. Line display common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_NONE for 1,8,11 PR_STANDARD for all other LineDisplay	PR_NONE for 11 PR_STANDARD otherwise
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	Not supported	
DataEventEnabled	Not supported	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	

Table 76. Line display common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	PN_DISABLED	OPOS_PN_DISABLED for 11, All for others
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 77. Line display specific properties

Property	JavaPOS and OPOS Gateway	OPOS
BlinkRate	Not supported	
CapBlink	Not supported except 10 (set CB_BLINKALL)	Not supported except 8 (set CB_BLINKALL)
CapBitmap	Not supported	
CapBlinkRate	Not supported	
CapBrightness	Not supported except 1 and 10 (Supported on EIA-232 integrated and external line displays)	
CapCharacterSet	See “Character sets supported by line display devices for JPOS” on page 104 and “Character sets supported by line display devices for OPOS” on page 106	All (values based on HW)
CapCursorType	See Table 81 on page 100.	Not supported
CapCustomGlyph	1, 2, 3, 6, 7, 8	Not supported
CapDescriptors	True for: USB 2x20 VFD External LineDisplay USB 2x20 VFD Integrated LineDisplay RS-485 2x20 VFD External LineDisplay USB Character/Graphic APA LineDisplay RS-485 APA LineDisplay False for the remaining of the LineDisplay devices	All: set true 1, 10, 11: set false
CapHMarquee	Not supported	

Table 77. Line display specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapICharWait	Not supported	
CapMapCharacterSet	Not supported	1
CapReadBack	Not supported	
CapReverse	Not supported	
CapScreenMode	8	Not supported
CapVMarquee	Not supported	
CharacterSet	All (values based on HW)	
CharacterSetList	All (values based on HW)	
CurrentWindow	All	
Columns	All	
CursorColumn	All	
CursorRow	All	
CursorType	See Table 81 on page 100. Not supported directly as property, but supported using DirectIO method for EIA-232 external LineDisplay (All points addressable (APA)).	Not supported
CursorUpdate	All	
CustomGlyphList	**1** "0015-001A,001C-001E" **2, 3, 6, 7** 0000-0007 **8** ASCII returns: ″0020-00FF″ English(437) returns: ″0001-0007,0009,000B,000C,000E-001A,001C-00FF″ Japanese(932) returns: ″8140-81FF,8240-84FF,8840-88FF,8940-9FFF,E040-E0FF,E140-EAFF″ Korean(1361) returns: ″A1A0-A1FF,A2A0-ACFF,B0A0-B0FF,B1A0-C8FF,CAA0-CAFF,CBA0-FDFF″ Simplified Chinese(936) returns: ″A1A0-A1FF,A2A0-A9FF,B0A0-B0FF,B1A0-F7FF″ Traditional Chinese(950) returns: ″A1A0-A1FF,A240-C6FF,C940-C9FF,CA40-F9FF″	Not supported

Table 77. Line display specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CustomDSGlyphList	Not supported	
CustomSSGlyphList	Not supported	
DeviceBrightness	Only 1	0 or 100% only except on 1 and 10 (0 to 100%)
DeviceColumns	All	
DeviceDescriptors	20 for: USB 2x20 VFD External LineDisplay USB 2x20 VFD Integrated LineDisplay RS-485 2x20 VFD External LineDisplay 24 for: USB Character/Graphic APA LineDisplay RS-485 APA LineDisplay	All
DeviceRows	All	
DeviceWindows	Not supported	All
GlyphHeight	**1, 2, 3** 7 **6, 7** 8 **8** 16 (2x20), 7 (4x20), 7 (5x20) **9** SBCS: 16 (2x20), DBCS: 16 (2x20)	Not supported
GlyphWidth	**1, 2, 3, 6, 7** 5 **8** 16 (2x20), 8(2x20), 5 (4x20), 5 (5x20) **9** SBCS: 8 (2x20), DBCS: 16 (2x20)	Not supported
InterCharacterWait	Not supported	
MapCharacterSet	Not supported	1
MarqueeFormat	Not supported	
MarqueeRepeatWait	Not supported	
MarqueeType	Not supported	
MarqueeUnitWait	Not supported	
MaximumX	Not supported	
MaximumY	Not supported	
Rows	All	

Table 77. Line display specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
ScreenMode	**1, 2, 3, 6, 7** 0 **8** 0, 1, 2 **9** 0	Not supported
ScreenModeList	**1, 2, 3, 6, 7** 2x20 **8** 2x20, 4x20, 5x20 **9** 2x20	Not supported

Table 78. Line display common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	Not supported	
clearOutput	Not supported	
close	All	
compareFirmwareVersion	Not supported	
directIO	7 command 1-ScreenMode, pData Value = (0-2x20,1-3x32)	**8** Command 1 = ScreenMode, pData Value = (0-2x20, 1-3x32) **10** Command 0 = ScreenMode, pDataValue = (2-2x20, 4-4x20, 5-5x20)
open	9 command 0 = ScreenMode pData Value = (2-2x20,4-4x20,5-5x20)	All
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 79. Line display specific methods

Method	JavaPOS and OPOS Gateway	OPOS
clearDescriptors	Supported for USB-VFD LineDisplay	All except 1 and 10
clearText	All	
createWindow	Not supported	All

Table 79. Line display specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
defineGlyph	1, 2, 3, 6, 7, 8 **Note: glyph** parameter is a byte array with length as follows: **1, 2, 3** 7 **6, 7** 8 **8** 32 (2x20), 16 (2x20), 7 (4x20), 7 (5x20)	Not supported
destroyWindow	Not supported	All
displayBitmap	Not supported	
displayText	All	
displayTextAt	All	
readCharacterAtCursor	Not supported	
refreshWindow	Not supported	All
scrollText	All	
setBitmap	Not supported	
setDescriptor	Supported only for USB-VFD LineDisplay	All except 1 and 10

Table 80. Line display events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All	
StatusUpdateEvent	All	

Table 81. CapCursor Type and Cursor Type Properties

Cursor type	2, 3, 4, 5, 6 (RS-485), 7 (RS-485), 9, 10, 11	6 (USB), 7 (USB)	1	8
capCursor Type	DISP_CCT_NONE	DISP_CCT_UNDERLINE +DISP_CCT_FIXED	DISP_CCT_UNDERLINE	DISP_CCT_UNDERLINE[1]
cursorType	DISP_CT_NONE	DISP_CT_UNDERLINE	DISP_CT_UNDERLINE, DISP_CT_NONE	DISP_CT_UNDERLINE[1] DISP_CT_NONE
[1]The 2x20 screen mode does not have a cursor.				

JavaPOS configuration

CharacterSetASCIIBehavior

The behavior of Character Set ASCII (998) is undefined for chars beyond 127 (0x7F). To make these chars behave as some other character set in the CharacterSetList, specify it using this property. To keep these chars working as previous JPOS releases, specify 998. The default if the property doesn't exist is 998.

```
="com.ibm.jpos.services.sdi.config.LineDisplay.CharacterSetASCIIBehavior" type="String" value="998"/
```

CursorState

```
<prop name="com.ibm.jpos.services.sdi.config.LineDisplay.CursorState" type="String" value="ON"/>
```

Initializes the EIA-232 LineDisplay VFD cursor state to On or Off. This is done only at initialization time. Valid values are ON or OFF.

lineDisplayId

```
<prop name="com.ibm.posj.bus.rs232.lineDisplayId" type="String" value="APA/>
```

For EIA-232 displays, this property is used to distinguish between All-Points Addressable (APA) and Vacuum Fluorescent Displays (VFD). Valid values are APA, VFD or LED

NumberOfRows

```
<prop name="com.ibm.jpos.services.sdi.config.LineDisplay.NumberOfRows" type="String" value="2"/>
```

Initializes the EIA-232 APA line display to a selected number of rows. Valid values are 2, 4, or 5.

JavaPOS DirectIO calls

The following DirectIO commands are supported for the LineDisplay. The syntax is as follows:

Syntax directIO (command: *int32*, **inout data:** *int32*, **inout obj:** *object*) **:**
void {raises-exception}

To access line display constants, import: `com.ibm.jpos.services.DirectIO`.

Select display text mode command

Note: This directIO call could be deprecated in future releases

Set display text mode. This command applies to RS485 and USB displays.

Table 82. Parameters for DirectIO.SELECT_PLUVFDIIAPA_DISPLAY_TEXT_MODE

Parameter	Type	Value
Command	Int32	DirectIO.SELECT_PLUVFDIIAPA_DISPLAY_TEXT_MODE
Data [0]	Int32	0 = display text in normal mode, 1 = display text rotated 180 degrees
Obj	Object	Always null

Select cursor type command

Set display cursor type. This command applies to EIA-232 displays.

Table 83. Parameters for DirectIO.SELECT_APA_VFD_CURSOR_TYPE

Parameter	Type	Value
Command	Int32	DirectIO.SELECT_APA_VFD_CURSOR_TYPE
Data [0]	Int32	4 = DISP_CT_UNDERLINE, 1 = DISP_CT_NONE

Table 83. Parameters for DirectIO.SELECT_APA_VFD_CURSOR_TYPE (continued)

Parameter	Type	Value
Obj	Object	Always null

Select number of rows command

Set number of display rows. This command applies to EIA-232 APA (all points addressable) character graphics displays.

This command can also be used with ScreenMode property.

Table 84. Parameters for DirectIO.SELECT_APA_NUMBER_OF_ROWS

Parameter	Type	Value
Command	Int32	DirectIO.SELECT_APA_NUMBER_OF_ROWS
Data [0]	Int32	2 = 2 rows by 20 columns, 4 = 4 rows by 20 columns, 5 = 5 rows by 20 columns
Obj	Object	Always null

Select rotate text 180 command

Set display test mode. This command applies to USB APA Vacuum Fluorescent Display (VFD).

Table 85. Parameters for the DirectIO.SELECT_ROTATE_TEXT_180

Parameter	Type	Value
Command	Int32	DirectIO.SELECT_ROTATE_TEXT_180
Data [0]	Int32	0 = display text in normal mode, 1 = display text mode 180 degrees rotated
Obj	Object	Always null

Errors A UposException may be thrown when this method is invoked. Some possible values of the exception ErrorCode properties are:

Table 86. ErrorCode property for DirectIO.SELECT_ROTATE_TEXT_180

Value	Possible causes
E_ILLEGAL	• cmd name /value is not known or is not valid. • Some of the parameter values are not correct.

Additional JavaPOS information

Descriptors

Most models of IBM vacuum fluorescent and character graphics displays have the ability to display descriptors. The following displays have 20 descriptors:

- USB 2x20 VFD External LineDisplay
- USB 2x20 VFD Integrated LineDisplay
- RS-485 2x20 VFD External LineDisplay
- All descriptors are on 1 row, below the bottom line of text.

The following displays have 24 descriptors:

- USB Character/Graphic APA LineDisplay
- RS-485 APA LineDisplay
- There are 2 rows of descriptors, 12 above the top row of text and 12 below the bottom row of text.

In prior versions of JavaPOS, the descriptors were numbered incrementally from 1, with position 1 being the top leftmost position. In JavaPOS 1.9.6, the descriptors have been renumbered to be consistent with OPOS. The descriptors are still numbered incrementally from 1, but the starting position is now the lowest rightmost position.

Capability definitions

DISP_CCT_NONE
Cursor is not displayable.

DISP_CCT_UNDERLINE
Cursor is displayable as an underline. Cursor can be turned on and off.

DISP_CCT_UNDERLINE+DISP_CCT_FIXED
Cursor is displayable as an underline. Cursor is always displayed.

Property definitions

DISP_CT_NONE
Cursor is not displayed.

DISP_CT_UNDERLINE
Cursor is displayed as an underline.

Note: Another value received in the device will cause a JPOS_E_ILLEGAL at setCursorType()

Character sets supported by line display devices for JPOS

Table 87 shows the country character sets supported by the line displays in JPOS. The US/European character set contains at least the upper case characters from many of the previous code pages (excluding Cyrillic). These are duplicated in the code pages for those countries.

Table 87. Character sets supported by line display devices in JPOS

Character set	LineDisplay device							
	APA (EIA-232)	VFD Integrated and Distributed EIA-232	VFD 1-Sided (USB)	LCD and 50-Key LineDisplay	APA (USB)	VFD 1-Sided and 2-Sided (RS-485)	APA (RS-485)	LED (RS-232)
101 (Mini-Alphanumeric)								✓
437 U.S. English	✓	✓	✓		✓	✓	✓	
998 US/European (based on CP437)	✓	✓	✓	✓	✓	✓		
808 (Cyrillic-Russia)		✓				Note 1		
819 (Alphanumeric and special)							Note 1	
852 (Central Europe or Hungary)		✓	✓		✓	Note 1		
855 (Cyrillic)		✓	✓		✓	Note1		
857 (Turkey)		✓	✓		✓	Note 1		
858 (Multilingual)		✓	✓		✓	Note 1		
862 (Israel)		✓	✓		✓	Note 1		
863 (Canada-French)		✓	✓		✓	Note 1		
864 (Arabic)		✓	✓		✓	Note 1		
865 (Nordic)		✓	✓		✓	Note1		
866 (Russia)			✓		✓	Note 1		
869 (Greek)		✓	✓		✓	Note1		
897 (Katakana)		✓	✓		✓	✓		
926 (Hangeul and Alphanumeric)							✓	
932 (Japanese)	✓				Note 3		Note 2	
936 (Simplified Chinese)	✓				3			
949 (Korean Wansung)					Note 3		Note 2	
950 (Traditional Chinese)	✓				Note 3		Note 2	
1361 (Korean Johab)	✓							

Notes:

1. Only available if EC Level >= 16 (0x10)
2. Only available if corresponding font is in device ROM
3. Only available if corresponding font was downloaded

OPOS configuration

Note: These settings cannot be modified with the Configuration tool.

Table 88. Service Object settings for RS-485/USB line display

Keyword	Type	Description	OPOS Gateway Support
DefineCharacter	Key	Key values for user-defined characters.	Yes
DefineCharacter\<X>	String	Hexadecimal value of user-defined character to be downloaded. Eight bytes of data for character definition. Each data byte is represented by 2 ASCII characters (IE, the value 0x69969960 is the string 0609090609090600). For more information on the character format, refer to *IBM Point of Sale Subsystem Programming Reference and User's Guide* (SC30-3560).	Yes

Table 89. Service object settings for EIA-232 line display

Keyword	Type	Description	OPOS Gateway Support
Type	String	Line display type. Valid values are: **LINE** Single-byte display **APA** Double-byte display	Yes
Mode	String	Screen display mode. Valid values are 2x20 (default), 4x20, and 5x20.	Yes

OPOS DirectIO calls

Set number of APA rows

Set number of rows on EIA-232 USB and RS-485 All-Points Addressable (APA) line displays.

Table 90. Set number of APA rows

Parameter	Type	Description
Command	Int32	0
Data	Long*	Pointer to a byte which determines the line display size. Valid values are: 2 (2x20) 3 (3x32) 4 (4x20) 5 (5x20)
Obj	BSTR*	Ignored

Additional OPOS information

USB APA font download

To download fonts to the USB Character/Graphics Display, you will need the following:

- The display font download program, AIPFNVFD.EXE, which comes with the IBM Point of Sale Subsystem for Windows.
- Font files for the USB Character/Graphics Display, which can be found in the IBM Point of Sale Subsystem for Windows BIN directory

Font File Name	Font File Description
jpnm16.bin	Japan, Mincho style
korm16.bin	Korea, Mincho style
sbcs0001.bin	SBCS fonts, updated version
Prcmz16.bin	PRC Simplified Chinese
Rocbg16b.bin	Taiwan Traditional Chinese

Downloading Fonts to the Display

Font files for the USB Character/Graphics Display must be downloaded manually using the AIPFNVFD.EXE program

```
AIPFNVFD -Sslot -Pport {-Ndevice-number} -Fx:\dir\font-file {-Q}
```

-Sslot The slot number (in decimal) of the display for which the download is intended.

-Pport The port number (in decimal) of the display for which the download is intended.

-Ndevice-number
An indication of the device number of the display for which the download is intended. Valid values are:

- A : device number 2A (default)
- B : device number 2B
- C : device number 2C
- D : device number 2D

-Fx:\dir\font-file
Fully qualified path name for the font file to be downloaded to the display.

-Q Quiet mode. No progress information is displayed

Character sets supported by line display devices for OPOS

Table 91 shows the country character sets supported by the line displays in 0POS. The US/European character set contains at least the upper case characters from many of the previous code pages (excluding Cyrillic). These are duplicated in the code pages for those countries.

Table 91. Character sets supported by line display devices in OPOS

Character set	LineDisplay device							
	APA (EIA-232)	VFD Integrated and Distributed EIA-232	VFD 1-Sided (USB)	LCD and 50-Key LineDisplay	APA (USB)	VFD 1-Sided and 2-Sided (RS-485)	APA (RS-485)	LED (RS-232)
101 (Mini-Alphanumeric)								✓
437 U.S. English	✓	✓	✓		✓	✓	✓	

Table 91. Character sets supported by line display devices in OPOS (continued)

Character set	LineDisplay device							
	APA (EIA-232)	VFD Integrated and Distributed EIA-232	VFD 1-Sided (USB)	LCD and 50-Key LineDisplay	APA (USB)	VFD 1-Sided and 2-Sided (RS-485)	APA (RS-485)	LED (RS-232)
998 US/European (based on CP437)	✓	✓	✓	✓	✓	✓		
808 (Cyrillic-Russia)								
819 (Alphanumeric and special)								
852 (Central Europe or Hungary)		✓	✓		✓	✓		
855 (Cyrillic)		✓	✓		✓	✓		
857 (Turkey)		✓	✓		✓	✓		
858 (Multilingual)		✓	✓		✓	✓		
862 (Israel)		✓	✓		✓	✓		
863 (Canada-French)		✓	✓		✓	✓		
864 (Arabic)		✓	✓		✓	✓		
865 (Nordic)		✓	✓		✓	✓		
866 (Russia)			✓		✓	✓		
869 (Greek)		✓	✓		✓	✓		
897 (Katakana)		✓	✓		✓	✓		
926 (Hangeul and Alphanumeric)								
932 (Japanese)	✓				✓			
936 (Simplified Chinese)								
949 (Korean Wansung)					✓			
950 (Traditional Chinese)								
1361 (Korean Johab)								

Chapter 12. Magnetic Ink Character Recognition

Supported devices

Table 92. Magnetic Ink Character Recognition supported devices

Device	Connectivity
4610 printer family MICR	EIA-232, RS-485, USB
Model 4 printer (OPOS only)	RS-485

Supported properties and methods

Table 93. Magnetic Ink Character Recognition common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	All	
DataEventEnabled	All	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 94. Magnetic Ink Character Recognition specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AccountNumber	All	
Amount	All	
BankNumber	All	Not supported
CapValidationDevice	All	
CheckType	All - cannot determine without exception processing	
CountryCode	All - cannot determine without exception processing	
EPC	All	
RawData	All	
ErrorReportingType	All	
SerialNumber	All	
TransitNumber	All	

Table 95. Magnetic Ink Character Recognition common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	All	
clearInputProperties	Supported	
clearOutput	Not supported	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 96. Magnetic Ink Character Recognition specific methods

Method	JavaPOS and OPOS Gateway	OPOS
beginInsertion	All	
beginRemoval	All	
endInsertion	All	
endRemoval	All	

Table 97. Magnetic Ink Character Recognition events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	

Table 97. Magnetic Ink Character Recognition events (continued)

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
ErrorEvent	All	
StatusUpdateEvent	All	

Magnetic Ink Character Recognition data

The format of the data read from the Magnetic Ink Character Recognition (MICR) looks like:

Transit Symbol	Transit Field 9-digit fixed	Transit Symbol	On-Us Field 19-character max	Amount Symbol	Amount Field 10-digit	Amount Symbol

Where:

Transit Field
9-digit number bracketed between the two Transit symbols. Bank Number is digits 5 through 8 of the Transit field.

On-Us Field
According to the X9.13 standard, section 8.4.1, the On-Us field ″usually contains the account number and *may* also contain a serial number and/or a transaction code.″ In addition, section 8.4.2 also states that ″*no recommendation is made* as to how the On-Us field is to be structured.″ This fact allows the individual financial institution the widest possible latitude in designing the field to suit its own internal system requirement. However, in no case shall the On-Us field contain more than 19 characters.

Amount Field
10-digit number bracketed between the two Amount symbols.

Auxiliary On-Us field
If it exists, it is to the left of the Routing Number (Transit field). It is also stated that ″when a serial number or other data is printed in this field, it must be preceded by, and followed by, an On-Us symbol″. The field length is variable up to 9 characters.

Only On-Us and Auxiliary On-Us fields need interpretation rules.

JavaPOS configuration

com.ibm.jpos.sdi.config.MICR.MinMicrSignalLevel

```
<prop name="com.ibm.jpos.sdi.config.MICR.MinMicrSignalLevel" type="String" value="50"/>
```

Defines a minimum signal level accepted for a successful MICR read. If a MICR reads below the minimum signal level an error event will be fired.

Valid values are 0 to 100. Default is 50.

com.ibm.jpos.sdi.config.MICR.stripAccountDashes

```
prop name="com.ibm.jpos.sdi.config.MICR.stripAccountDashes" type="String" value="false"/
```

Determines whether to remove any dashes from the account number. When true, a value of ″1234-56-7″ would become ″1234567″.

Valid values are TRUE or FALSE.

com.ibm.jpos.sdi.config.MICR.stripTransitDashes

```
<prop name="com.ibm.jpos.sdi.config.MICR.stripTransitDashes" type="String" value="false"/>
```

Determines whether to remove any dashes from the transit number. When true, a value of ″3579-12-″ would become ″357912″.

Valid values are TRUE or FALSE.

com.ibm.jpos.sdi.config.MICR.switchTransitDashToSpace

```
<prop name="com.ibm.jpos.sdi.config.MICR.switchTransitDashToSpace" type="String" value="false"/>
```

Determines whether to change any dashes in the transit number to spaces. When true, a value of ″12-4-66-7″ would become ″12 4 66 7″.

Valid values are TRUE or FALSE.

Additional JavaPOS information

MICR exception tables

An exception table contains Transit Field/On-Us Field interpretation rule pairs. This enables you to manually code the On-Us field interpretation rule based on the Transit Field because each bank is likely to have a different Transit Field. For this purpose, three formats for Exception Table are supported. The first format (format 1) is an XML description of how the data is to be extracted from the check, and should be written in a separate file. Formats two (2) and three (3) can be put directly into the MICR JposEntry.

Format 1 (supported since JavaPOS 1.7.5)

This is an XML representation of the On-Us and Auxiliary On-Us fields stored in an independent file. This file is identified in a property in the JposEntry for the MICR device:

```
<prop name="com.ibm.jpos.sdi.config.MICR.exceptionTableFile"
   Type="String"
   Value="<file-path-goes-here>"/>
```

If this property is omitted JavaPOS will search the classpath for a file named `ibmmicrexception.xml`.

This file can also contain entries for legacy formats (PositionSkip and Symbol formats). An entry for legacy format must be a child of 'MICRExceptionTable' and is just like the `prop` elements described for formats 2 and 3.

For example:

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

<MICRExceptionTable FormatToUse="XML">
 <ExceptionEntry TransitField="123456789" CheckType="Business">
  <OnUsFieldParsingRule>
   <SkipCharacterLength value="1" />
   <AccountNumberLength value="5" />
   <SkipCharacterLength value="1" />
   <SerialNumberLength value="5" />
  </OnUsFieldParsingRule>
  <AuxOnUsFieldParsingRule>
   <SerialNumberLength value="5" />
   </AuxOnUsFieldParsingRule>
  </ExceptionEntry>
 <prop name="com.ibm.jpos.sdi.config.MICR.exceptionTable0"
    type="String"
    value="P123456789AAAAAAAAAAXSSSSSSSS"/>
 <prop name="com.ibm.jpos.sdi.config.MICR.exceptionTable1"
    type="String"
    value="B778899001D154R"/>
</MICRExceptionTable>
```

A sample file called `ibmmicrexception.xml.Sample` is included with the installation. This can be useful as a starting point for configuring your own exception tables. The location for `ibmmicrexception.xml.Sample` is:

- For Windows: C:\POS\IBMJPOS\Docs
- For Linux: /opt/ibm/javapos/docs

The following considerations must be applied:

- There can be as many ExceptionEntry and prop elements as needed. ExceptionEntry elements must come before any prop element.
- Values for transit number (TransitField in ExceptionEntry and characters 2-10 in the case of prop elements) must be unique.
- Valid values for the attribute FormatToUse are XML and Legacy. The default is XML. This indicates which format is to be read first. Since any repeated value for transit number is dropped, this switches priority from one format to the other.
- Formats 2 and 3 (legacy formats) are read in order of the property names, first the ones in the new ibmmicrexception.xml and then the ones in the JposEntry. All listed entries must have consecutive names inside each file, and counts should start at 0.
- Order of elements within OnUsFieldParsingRule is strict. The order shown above is the expected order. OnUsFieldParsingRule is required.
- SerialNumberLength and AccountNumberLength can be in either order, and SkipCharacterLength can be at the beginning or between SerialNumberLength and AccountNumberLength.
- Exactly one occurrence of AccountNumberLength is required.
- SerialNumberLength and SkipCharacterLength are optional.
- All *value* attributes are required. TransitField attribute from ExceptionEntry element is also required.
- CheckType attribute from ExceptionEntry is optional. If specified, it must be either *Personal* or *Business*. The default is *Personal*.
- The serial number may appear in the Auxiliary On-Us field; AuxOnUsFieldParsingRule holds a SerialNumberLength element if this is the case. AuxOnUsFieldParsingRule is optional.

- SerialNumberLength should appear in one field only. If it appears in both, the one from AuxOnUsFieldParsingRule is ignored.

Format 2 (supported since JavaPOS 1.7.2)

The On-Us field interpretation rule is as follows:

A account number character (can be a dash)

S serial number character (can be a dash)

X ignored (no parsing)

A typical MICR exception table looks like this:

```
<prop name="com.ibm.jpos.sdi.config.MICR.exceptionTable0"
    type="String" value="P123456789AAAAAAAAAAXSSSSSSSS"/>
<prop name="com.ibm.jpos.sdi.config.MICR.exceptionTable1"
    type="String" value="B123456709AAAAAAAAXXXSSSSSSSS"/>
```

where the value attribute is parsed as shown in Table 98.

Table 98. MICR exception table fields for format 2

Characters	Content
1	Check type: B = Business check P = Personal check
2 to 10	Transit field.
11 to end	On-Us Field. The A,X,S field is not required to be 19 characters long. You can choose the number of A, X, or S characters to encode the rule. This rule is applicable to the first valid digit immediately after the Transit Field. The On-Us symbol can be part of the On-Us Field, and the parser is expected to handle this condition.

Format 3 (supported since JavaPOS 1.5.1)

Exception configuration properties are provided for MICR processing to handle MICR data that do not conform to some of the standard formats. Default exception tables are set up for each MICR entry in the file JPOS.xml.

Use the following format:

```
<prop name="exceptionTable0" type="string" value= "B778899001D154R"/>
```

where each byte represents the following:

P/B (byte 1)
Type of check: either *Business* or *Personal*. This option is not used for parsing but instead is used to set the check type property, which is usually set to UNKNOWN.

TRANSIT (byte 2-10)
A 9-digit bank transit number that is key for determining if the exception table entry is applied to the MICR data.

SPC (byte 11)
SPC options:

- If SPC is set to *D*, all spaces are removed from the primary On-Us field for processing
- If SPC is not set to *D*, the On-Us field remains unchanged.

Field (byte 12)
Not used. The driver always use the first field in the primary On-Us field.

SKIP (byte 13)
The account starts in this index and includes the remaining characters to the right in the primary On-Us field. (The index count starts at 0.)

SIZE (byte 14)
Not used because the driver identifies the serial number.

ACC (byte 15)
Not used.

The following table shows sample output for Format 3:

Table 99. Format 3 Exception Table (sample data)

Raw Data	t123456780t 0123456789o 0671	
Primary On-Us	0123456789o 0671	
Field #1	0123456789	
Exception Table entry	**Account**	**Serial**
P123456780D111R	123456789	0671
P123456780D165R	6789	0671
P123456780D295R	9	0671
P123456780D422R	23456789	0671

OPOS configuration

Table 100. Service object settings for MICR

Keyword	Type	Description	OPOS Gateway Support
ExceptionFile	String	Fully-qualified filename of MICR exception file, which contains exception processing information. The default value is: C:\POS\BIN\PARSE.DAT	Yes
MinMicrSignalLevel	String	Minimum signal string (0-100) for successful MICR read. The default value is 50 (full string).	Yes
SlipRegTimeout	String	Time (in milliseconds) to wait for check to be registered. **Note:** This setting cannot be modified with the Configuration tool.	No
SlipRegMinDelay	String	Time (in milliseconds) to delay before sending document register command. **Note:** This setting cannot be modified with the Configuration tool.	No

OPOS DirectIO calls

Add new MICR exception

Adds the specified MICR exception string into the current MICR exception file.

Table 101. Add new MICR exception

Parameter	Type	Description
Command	Int32	101
Data	Long*	Ignored

Table 101. Add new MICR exception (continued)

Parameter	Type	Description
Obj	BSTR*	Pointer to a string containing the MICR exception string. See “Additional OPOS information” for more information on the format of the exception string.

Additional OPOS information

MICR exception file

Exception configuration properties for MICR processing enable you to handle MICR data that do not conform to some of the standard formats. Default exception tables are set up for each MICR entry in the file <INSTALL_DIR>\bin\parse.dat.

Use the following format to create exception table entries:

```
#P/B TRANSIT  SPC FIELD SKIP SZ ACC
  P  123456789 D  1  5  5  R
```

Each parameter must be separated for any number of spaces, and each byte represents the following:

P/B (byte 1)
The type of check: either "Business" or "Personal." This option is not used for parsing but instead is used to set the check type property, which is usually set to "UNKNOWN."

TRANSIT (byte 2-10)
9-digit bank transit number, which is used as the key to determine if the exception table entry is applied to the MICR data.

SPC (byte 11)
If SPC is set to *D*, all spaces are removed from the primary On-Us field for processing; otherwise, the On-Us field remains unchanged.

Field (byte 12)
Indicates which subfield in the primary On-Us field is used to get the account number and the serial number.

SKIP (byte 13)
The account starts in this index and includes the remaining characters to the right in the primary On-Us field. (The index count starts at 0.)

SIZE (byte 14)
The length of the serial number.

ACC (byte 15)
The alignment of the account number in the subfield:

- If it is "L," then the account number aligns on the left of the subfield.
- If it is "R," then the account number aligns on the right of the subfield.

The following table shows sample output:

Table 102. Exception configuration properties for MICR processing: Example 1

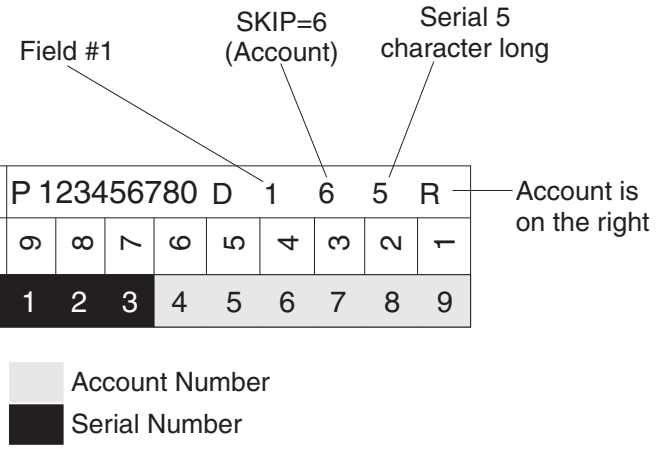
Raw Data	t123456780t 0123456789o 0671
Primary On-Us	0123456789o 0671
Field #1	0123456789

Table 102. Exception configuration properties for MICR processing: Example 1 (continued)

Field #2	0671	
Exception Table entry	**Account**	**Serial**
P123456780D 1 1 1 R	9	8
P123456780D 1 6 5 R	456789	0123
P123456780D 2 1 5 R	1	067
P123456780D 4 1 5 R	1	067
P123456780D 2 2 2 L		06
P123456780D 1 1 1 L	01234567	8
P123456780D 1 3 5 L	01	23456

The following image illustrates how the information is parsed to retrieve the Account and Serial numbers for the table entry "P123456780D 1 6 5 R":

Figure 20. Data parsing: Example 1



The following image illustrates how the information is parsed to retrieve the Account and Serial numbers for the table entry "P123456780D 1 3 5 L":

Figure 21. Data parsing: Example 2

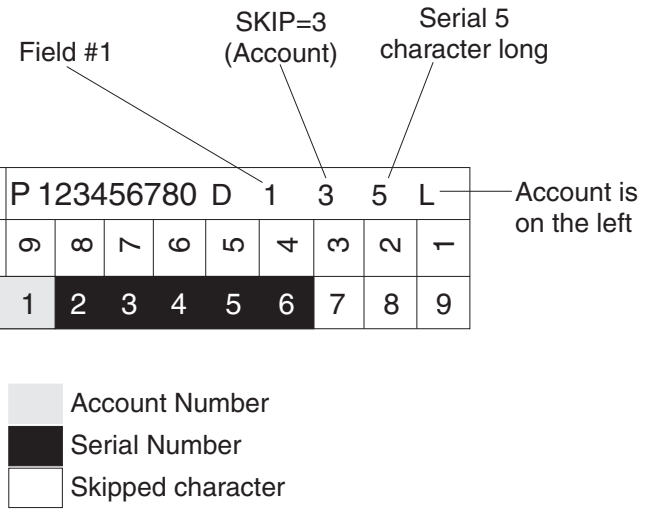


Table 103. Exception configuration properties for MICR processing: Example 2

Raw Data	o87654321o t123456780t 0123456789o
Primary On-Us	0123456789o

Table 103. Exception configuration properties for MICR processing: Example 2 (continued)

Field #1	0123456789	
Exception Table entry	**Account**	**Serial**
P123456780D 1 6 5 R	456789	0123
P123456780D 4 1 5 R	9	45678
P123456780D 2 2 2 L	012345	67
P123456780D 1 3 5 L	01	23456

Chapter 13. Motion sensor

Motion sensor supported devices

Table 104. Motion sensor supported devices

Device	Connectivity
1. SurePOS 500/600	Integrated
2. Kiosk	Integrated

Supported properties and methods

Table 105. Motion sensor common properties

Property	JavaPOS and OPOS Gateway	OPOS
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_NONE	PR_NONE
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
PowerNotify	All	
PowerState	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 106. Motion sensor specific properties

Property	JavaPOS and OPOS Gateway	OPOS
Motion	All	

Table 106. Motion sensor specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
Timeout	All	

Table 107. Motion sensor common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	Always shareable	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	Always shareable	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 108. Motion sensor specific methods

Method	JavaPOS and OPOS Gateway	OPOS
waitForMotion	All	

Table 109. Motion sensor events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
StatusUpdateEvent	All	

Additional JavaPOS information

This section describes how to load the correct motion sensor driver in Linux for your hardware.

Add the following lines in /opt/ibm/javapos/etc/machine.conf:

```
<keyword> <number of entries>
<machine type><model number>
```

″<keyword>″ represents the device for which a driver is needed and ″<number of entries>″ indicates the number of ″<machine type><model number>″ pairs that follow. If the system on which JavaPOS is running matches one of the specified ″<machine type><model number>″ pairs, the device driver for that ″<machine type><model number>″ will be loaded.

For the motion sensor, machine.conf might have the following entries:

```
Motion 2
4838132
4838135
```

In this case the <number of entries> value is **2**.

Chapter 14. Magnetic stripe reader

Supported devices

Table 110. Magnetic stripe reader (MSR) supported devices

Device		Connectivity
ISO (3-track)	1. Retail alphanumeric POS keyboard w/Card Reader	PS/2, RS-485, USB
ISO (3-track)	2. Retail alphanumeric POS keyboard with MSR and pointing device	PS/2
ISO (3-track)	3. Retail POS keyboard with card reader	RS-485, USB
ISO (3-track)	4. Retail POS keyboard with card reader and display	RS-485, USB
ISO (3-track)	5. Modifiable layout keyboard with card reader	RS-485, USB
ISO (3-track)	6. 32-key programmable keypad with ISO MSR	EIA-232, RS-485, USB
JUCC	7. 32-key programmable keypad with JUCC MSR	EIA-232, RS-485, USB
ISO (3-track)	8. SurePoint 4820 integrated ISO MSR	EIA-232, RS-485, USB
JUCC	9. SurePoint 4820 integrated JUCC MSR	EIA-232, RS-485, USB
ISO (3-track)	10. SurePOS 500/600 integrated ISO MSR	EIA-232
JUCC	11. SurePOS 500/600 integrated JUCC MSR	EIA-232
ISO (3-Track)	12. Anyplace Kiosk integrated ISO MSR	EIA-232
JUCC	13. Anyplace Kiosk integrated JUCC MSR	EIA-232
ISO (3-Track)	14. SureOne Integrated ISO MSR	PS/2
ISO (3-Track)	15. . Compact Alphanumeric POS Keyboard (requires manual firmware update for support)	PS/2
JUCC	16. POS keyboard V	RS-485, USB
JUCC	17. PC POS Keyboard or PC Point of Sale Keyboard	PS/2, RS-485, USB
JUCC	18. Retail POS keyboard VI	RS-485, USB
ISO (3-Track)	19. SurePOS 100 integrated ISO MSR	PS/2
JUCC	20. 4674 Integrated JUCC MSR	RS-485
JUCC	21. 4685-KC1 (OPOS only)	RS-485
JUCC	22. 4685-K01 (OPOS only)	RS-485
JUCC	23. 4685–K02 (Ultra7) keyboard with card reader	RS-485, USB
JUCC	24. 4685-K02 with MSR/Encoder and 4 position keylock	RS-485
JUCC	25. 4685-K02 with MSR/Encoder and 6 position keylock	RS-485
JUCC	26. 4685-K03	RS-485, USB
ISO (3-Track)	27. Modular alphanumeric keyboard	PS/2, USB
JUCC	28. Modular alphanumeric keyboard	PS/2, USB
ISO (3-Track)	29. Modular Compact Alphanumeric keyboard	PS/2, USB
JUCC	30. Modular Compact Alphanumeric keyboard	PS/2, USB
ISO (3-Track)	31. Modular 67 Key Keyboard	PS/2, USB
JUCC	32. Modular 67 Key Keyboard	PS/2, USB

Supported properties and methods

Table 111. MSR common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_NONE for EIA-232 and PS/2 PR_STANDARD for all other MSR devices	PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	All	
DataEventEnabled	All	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 112. MSR specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AccountNumber	All	
CapISO	All except JUCC 4840 MSR	All except JUCC 4840 MSR
CapJISOne	All	All except 4840 MSR, SureOne Built-In, and 4820 MSR
CapJISTwo	All	All except 4840 MSR and SureOne Built-In

Table 112. MSR specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapTransmitSentinels	False	All
DecodeData	All	
ErrorReportingType	All	
ExpirationDate	All	
FirstName	All	
MiddleInitial	All	
ParseDecodeData	All	
ServiceCode	All	
Suffix	All	
Surname	All	
Title	All	
Track1Data	All	
Track1DiscretionaryData	All	
Track2Data	All	
Track2DiscretionaryData	All	
Track3Data	All	
Track4Data	All JUCC	
TracksToRead	All	
TransmitSentinels	Not supported	

Table 113. MSR common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	All	
clearOutput	Not supported	
close	All	
compareFirmwareVersion	Not supported	
directIO	RS-485 – 4685-K02 (Ultra 7)	Not supported
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 114. MSR events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	

Table 114. MSR events (continued)

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
ErrorEvent	All	
StatusUpdateEvent	All	

JavaPOS configuration

enableOnlineWatcher

•

```
<prop name="com.ibm.posj.bus.rs232.enableOnlineWatcher" type="Boolean" value="true"/>
```

Monitors the device for being online or offline and performs appropriate actions. To configure the poll time for events, modify the com.ibm.posj.bus.rs232.onLineWatcherPollTime.MSR property in the posj.properties file.

Values accepted

- TRUE - Default value
- FALSE

Device Support

- EIA-232 : 12, 13

msrID

```
<prop name="com.ibm.posj.bus.rs232.msrId" type="String" value="ISO"/>
```

Determines whether the EIA-32 MSR hardware is a 3-track ISO unit or a double-sided JUCC unit. Valid values are ISO or JUCC. Default is ISO.

Values accepted

- ISO- Default value
- JUCC

Device Support

- EIA-232 : 6, 7, 8, 9, 10, 11, 12, 13

JavaPOS DirectIO calls

The following DirectIO commands are supported for the MSR. The syntax is as follows:

Syntax directIO (command: *int32*, **inout data:** *int32*, **inout obj:** *object*) **: void {raises-exception, use after open-claim-enable}**

MSR write data command

Note: This command is used to write data to the JIS magnetic stripe of payment cards conforming to the JIS specification.

See Table 115 for a reference of the parameters to call DirectIO.MSR_WRITE_DATA_CMD.

Table 115. Parameters for DirectIO.MSR_WRITE_DATA_CMD

Parameter	Type	Value
Command	Int32	DirectIO.MSR_WRITE_DATA_CMD
Data [0]	Int32	Write wait timeout value in milliseconds. The wait timeout can be set to JposConst.JPOS_FOREVER and the command has no timeout to return. The maximum wait timeout is 25,000 milliseconds (25 seconds).
Obj	Object	String representation of the data to be written. The maximum length for the object is 69 characters. The first 69 characters will be written to the magnetic stripe and additional characters will be ignored.

Remarks

A card with a JIS II track must be swiped through the MSR during the directIO call with MSR_WRITE_DATA_CMD so data can be stored. Otherwise an exception is thrown.

Device Support

- 24, 25

Errors

A UposException may be thrown when this method is invoked. Some possible values of the exception's ErrorCode property are:

Table 116. UposException error code property

Value	Possible causes
E_ILLEGAL	One of the following errors has occurred: • The device is not write enabled. • Some of the parameter values are not correct. • Command sent to not capable write MSR
E_TIMEOUT	Timeout expired and card was not swiped.

Additional JavaPOS information

SureOne MSR

SureOne firmware discards any invalid MSR reading and sends only valid track data. Track errors are not reported.

If track 2 is empty and track 3 contains data, the track 3 data is displayed as track 2 and no data appears in track 3.

OPOS configuration

Table 117. Service object settings for EIA-232 MSR

Keyword	Type	Description	OPOS Gateway Support
MsrType	String	Type of MSR reader attached. Valid values are: **ISO** Standard three-track reader **JUCC** JIS-I and II reader	Yes

Additional OPOS information

SureOne/SurePOS 100 MSR limitations

The MSR on the SureOne is part of the Keyboard. The raw data is in ASCII format. When a credit card is swiped with the MSR DataEventEnabled set to *false*, the incoming data is treated as keyboard data. The only way to get MSR data events is to set DataEventEnabled to *true*. In order to queue MSR data, both DataEventEnabled and FreezeEvents must be set to *true*.

Chapter 15. POS keyboard

POS keyboard supported devices

Table 118. POS keyboard supported devices

Device	Connectivity
1. Retail alphanumeric POS keyboard with card reader Also known as: ANPOS or NANPOS	PS/2, RS-485, USB
2. Retail alphanumeric POS keyboard with MSR and pointing device Also known as: ANPOS with pointer	PS/2
3. Retail POS keyboard Also known as: 50-Key keyboard	RS-485, USB
4. Retail POS Keyboard with card reader Also known as: 50-Key/MSR	RS-485, USB
5. Retail POS Keyboard with card reader and display Also known as: 50-Key/Display	RS-485, USB,
6. Modifiable Layout Keyboard with card reader Also known as: 133 Key or Matrix	RS-485, USB
7. 32-Key programmable keypad with MSR Also known as: 4820 Keyboard	RS-485, USB, PS/2
8. SureOne Integrated Keyboard	PS/2
9. Compact Alphanumeric POS Keyboard Also known as: CANPOS	PS/2
10. POS Keyboard V	RS-485, USB
11. PC POS Keyboard or PC Point of Sale keyboard Also known as: ANKPOS	PS/2, RS-485, USB
12. PLU Keyboard/Display III	RS-485, USB
13. Retail POS Keyboard VI Also known as: 50-Key keyboard with JUCC MSR	RS-485, USB
14. SurePOS 100 Integrated keyboard	PS/2
15. 4674 Integrated keyboard	RS-485
16. 4685-KCI (OPOS only)	RS-485
17. 4685-K01 (OPOS only)	RS-485
18. 4685-K02 (Ultra 7) keyboard with card reader	RS-485, USB
19. 4685-K02 with MSR/Encoder and 4 position keylock	USB
20. 4685-K02 with MSR/Encoder and 6 position keylock	USB
21. 4685-K03	RS-485, USB
22. Modular Alphanumeric keyboard	PS/2, USB
23. Modular Compact Alphanumeric Keyboard	PS/2, USB

Table 118. POS keyboard supported devices (continued)

Device	Connectivity
24. Modular 67 Key Keyboard	PS/2, USB

Supported properties and methods

Table 119. POS keyboard common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	All	
DataEventEnabled	All	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 120. POS keyboard specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapKeyUp	All	
EventTypes	All	

Table 120. POS keyboard specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
POSKeyData	All	
POSKeyEventType	All	

Table 121. POS keyboard common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	All	
clearOutput	All	
close	All	
compareFirmwareVersion	Not supported	
directIO	All	Not supported
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 122. POS keyboard events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	
DirectIOEvent	All	Not supported
ErrorEvent	All	
StatusUpdateEvent	All	

CANPOS keyboard

The CANPOS keyboard firmware version must be at least 1.2.3 for the CANPOS automatic download utility to work. The firmware can be automatically updated to newer versions in UPOS. The update, if required, typically takes three to five minutes, depending on operating system.

The keyboard function keys can also be updated automatically, but you must create your own keyboard configuration file. To create a configuration file, a separate utility is available from the IBM Retail Store Solution support Web site. To download and use this utility, perform the following steps:

1. Go to http://www.ibm.com/solutions/retail/store/support.
2. Click **Other Systems and Devices**.
3. Click **Firmware and Utilities for CANPOS**. All utilities are packaged in a single zip file for supporting different operating systems. Based on your preferred system, you can download the appropriate utility.

4. Unzip the package and choose the appropriate utility to generate the configuration file.
5. Rename or save your file as aipcnpos.pcf and place it in folder \pos\bin for Windows systems, and /usr/share/pos for Linux systems.
6. Reboot your POS system and the configuration file is loaded automatically.

To confirm the success of the firmware and configuration download, you can check aipanpos.log under \pos\log in Windows, or /var/log in Linux POS systems.

Installing USB System attached keyboard on Windows XP

The proper device driver is automatically installed for the USB System keyboard on Windows XP, if you select **IBM Alphanumeric Point of Sale Keyboard** and indicate that it is attached to a USB port during installation. After reboot, a Windows message is displayed, indicating that you are installing the POS USB Keyboard. Click **Continue Anyway** to install IBM's driver after seeing this message.

Note: If you are installing the keyboard for the first time, the system must be rebooted again before the new driver can be used.

JavaPOS configuration

Click

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.Click" type="Byte" value="0x00"/>
```

JavaPOS provides a user-configurable mechanism to modify the loudness of the keyboard click. This is accomplished by specifying the following in the keyboard's JposEntry

Valid values for the Click property are listed below. The default is 0x00.
0x00 (off)
0x01 (soft)
0x02 (loud)

Device Support:

- 1, 2, 3, 4, 5, 6, 7, 11, 13, 19, 20, 21, 22, 23, 24
- USB Only: 10, 12, 18, 21

*USB not supported when configured as system keyboard

FatFingerTimeout

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.FatFingerTimeout" type="Byte" value="0x03"/>
```

A fat-finger situation occurs when an operator attempts to press a single key, but misses the center of the key, and presses both the intended key and an adjacent key. The time interval between subsequent key strokes is measured by the keyboard's processor. If the two keys were pressed within a certain interval, a fat-finger situation is assumed, and a status indicating this occurrence accompanies the scan codes to the terminal.

Device Support:

- 1 (RS-485, USB)*, 3, 4, 5, 6, 7 (RS-485, USB), 11 (RS-485, USB)*, 13, 19, 20, 24
- USB Only: 10, 12, 18, 21, 22*, 23*

*USB not supported when configured as system keyboard

Valid values are 0 through 4. The default is 3.
0 = 0 milliseconds (fat-finger detection disabled)
1 = 10 milliseconds
2 = 20 milliseconds
3 = 30 milliseconds
4 = 40 milliseconds

ExtendedKeyMapping

```
<prop name="com.ibm.jpos.sdi.config.POSKeyboard.ExtendedKeyMapping" type="Boolean" value="true"/>
```

The extended scan code property is used to differentiate between two keys having the same hardware scan code. A typical extended scan code is represented by two scan codes, for example: X'e0' X'23'.

When the ExtendedKeyMapping property is true, the first scan code of the extended key is replaced by X'01'; the second scan code is not altered. For example, the extended key X'e0' X'23' is returned to the application as X'123'.

When the ExtendedKeyMapping property is false, the extended keys are not supported. Only a single scan code is returned to the application (X'0023').

Valid values are TRUE and FALSE. The default is TRUE.

Device Support:

- PS2 : 1, 2, 8, 9, 11, 14, 22, 23
- USB when configured as system keyboard : 1, 11, 22, 23

KbdScanning

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.KbdScanning" type="Boolean" value="true"/>
```

This property controls whether to enable or disable keyboard scanning. When scanning is enabled the keyboard generates make and break scan codes for each key press.

Device Support:

- 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 18, 19, 20, 21, 22, 23, 24

Valid values are TRUE and FALSE. The default is TRUE.

KeyTranslationFile

```
<prop name="KeyTranslationFile" type="String" value="\keys.dat"/>
```

The KeyTranslationFile property contains the full path name of the file containing key translation information. A sample file called **keys.dat** is included with the installation.

The location of the sample file will depend on your operating system.

Linux - /opt/ibm/javapos/docs

Windows - C:\POS\IBMJPOS\Docs

Typematic

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.Typematic" type="Boolean" value="true"/>
```

This property controls the typematic function of the keyboard.

Most of the keys on all keyboards are optionally typematic keys. You can enable or disable the typematic function for the entire keyboard with this option.

If the typematic function is disabled, and a key is pressed, then the make key code is generated only once, no matter how long that key is held down. If the key is released the break code is sent.

If the typematic function is enabled and you press a key, the make key code is generated. If the key remains pressed for a period longer than the amount of time defined by typematic delay in the jpos.xml, another make key code is generated. As long a the key remains pressed, make key codes are generated at the rate defined by typematic rate until the key is released, at which time the break code is sent.

Valid values are TRUE and FALSE. The default is TRUE.

Device Support

- 1*, 2, 3, 4, 5, 6, 7, 11*, 13, 19, 20, 21
- USB Only: 10, 12, 18, 21, 22*, 23*, 24

* USB on Linux not supported when configured as system keyboard.

TypematicDelay

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.TypematicDelay" type="Byte" value="0x01"/>
```

Device Support:

- 1 (RS-485, USB)*, 3, 4, 5, 6, 7 (RS-485, USB), 11 (RS-485, USB)*, 13, 19, 20, 22*,23*
- USB Only: 10, 12, 18, 21

*USB not supported when configured as system keyboard

The typematic delay determines how long the key must be pressed before the typematic function is enabled.

Valid values for the TypematicDelay property are listed below. The default is 0x01.

0x00 = Delay 250 milliseconds
0x01 = Delay 500 milliseconds
0x02 = Delay 750 milliseconds
0x03 = Delay 1000 milliseconds

TypematicRate

The typematic rate determines how many make codes are generated per second (+/-20%).

```
<prop name="com.ibm.jpos.sdi.config POSKeyboard.TypematicRate" type="Byte" value="0x14"/>
```

Device Support:

- 1 (RS-485, USB)*, 3, 4, 5, 6, 7 (RS-485, USB), 11 (RS-485, USB)*, 13, 19, 20, 22*, 23*
- USB Only: 10, 12, 18, 21,

*USB not supported when configured as system keyboard

Valid values for the TypematicRate property are listed below. The default is 0x14.

0x00 = Rate 2.0
0x01 = Rate 2.1
0x02 = Rate 2.3
0x03 = Rate 2.5
0x09 = Rate 4.3
0x0a = Rate 4.6
0x0b = Rate 5.0
0x0c = Rate 5.5
0x0d = Rate 6.0
0x0e = Rate 6.7
0x0f = Rate 7.5
0x10 = Rate 8.0
0x11 = Rate 8.6
0x12 = Rate 9.2
0x13 = Rate 10.0
0x14 = Rate 10.9
0x15 = Rate 12.0
0x16 = Rate 13.3
0x17 = Rate 15.0
0x18 = Rate 16.0
0x19 = Rate 17.1
0x1a = Rate 18.5
0x1b = Rate 20.0
0x1c = Rate 21.8
0x1d = Rate 24.0
0x1e = Rate 26.7
0x1f = Rate 30.0

JavaPOS DirectIO calls

Syntax directIO (command: *int32*, **inout data:** *int32*, **inout obj:** *object* **): void { raises-exception, use after open-claim-enable}**

To access DirectIO constants import com.ibm.jpos.services.DirectIO.

Get keyboard light status command

This command returns the indicators status in data[0].

Table 123. DirectIO.GET_KBD_LIGHT_STATUS_CMD

Parameter	Type	Description
Command	Int32	DirectIO.GET_KBD_LIGHT_STATUS_CMD
Data	Int32	An `int` array with at least one element
Obj	Object	Any value. A null value is accepted

Remarks

Returns the indicators status on data[0] reference parameter. The bit values in the int parameter (data[0]) are:

- '1' = LED is to be turned on
- '0' = LED is to be turned off

Table 124. DirectIO.GET_KBD_LIGHT_STATUS_CMD bit representation

Parameter	Bit representation of the data element	
7–31 bits	Reserved - must	= 0
6	SCROLL-LOCK	= DirectIO.KBD_SCROLL_LOCK_LIGHT
5	CAPS-LOCK	= DirectIO.KBD_CAPS_LOCK_LIGHT
4	NUM-LOCK	= DirectIO.KBD_NUM_LOCK_LIGHT
3	"no label" or READY	= DirectIO.KBD NO_LABEL_LIGHT
2	MSG PEND or SYS MSG	= DirectIO.KBD_MESSAGE_PENDING_LIGHT
1	OFFLINE	= DirectIO.KBD_OFFLINE_LIGHT
0	WAIT	= DirectIO.KBD_WAIT_LIGHT

Set keyboard light on command

This command illuminates the indicator specified on the data[0] parameter.

Table 125. DirectIO.TURN_KBD_LIGHT_ON_CMD bit representation

Parameter	Type	Description
Command	Int32	DirectIO.TURN_KBD_LIGHT_ON_CMD
Data[0]	Int32	Data can be set to any of the following options: DirectIO.KBD_WAIT_LIGHT DirectIO.KBD_OFFLINE_LIGHT DirectIO.KBD_MESSAGE_PENDING_LIGHT DirectIO.KBD_NO_LABEL_LIGHT DirectIO.KBD_NUM_LOCK_LIGHT DirectIO.KBD_CAPS_LOCK_LIGHT DirectIO.KBD_SCROLL_LOCK_LIGHT
Obj	Object	Any value. A null value is accepted.

Set keyboard light off command

This command turns off the indicator specified on the data[0] parameter.

Table 126. DirectIO.TURN_KBD_LIGHT_OFF_CMD

Parameter	Type	Description
Command	Int32	DirectIO.TURN_KBD_LIGHT_OFF_CMD
Data[0]	Int32	Data can be set to any of the following options: DirectIO.KBD_WAIT_LIGHT DirectIO.KBD_OFFLINE_LIGHT DirectIO.KBD_MESSAGE_PENDING_LIGHT DirectIO.KBD_NO_LABEL_LIGHT DirectIO.KBD_NUM_LOCK_LIGHT DirectIO.KBD_CAPS_LOCK_LIGHT DirectIO.KBD_SCROLL_LOCK_LIGHT
Obj	Object	Any value. A null value is accepted.

A UposException might be thrown when this method is invoked. Some possible values of the exception ErrorCode property are:

Additional JavaPOS information

Configuration download for Modular POS Keyboards

Supported Devices: 22, 23 and 24 (USB Only) The IBM JavaPOS Printer driver provides a mechanism for downloading configuration files to the Modular POS Keyboards using the Update Device Firmware tool. Follow the steps listed below to configure the devices using JavaPOS:

1. Generate the configuration file using the ‘Modular POS Keyboard Firmware & Configuration’ utility.

   Note: The utility can be downloaded from IBM support web site.
2. Copy the configuration file to the JavaPOS directory

   Windows : “\config” directory
   Linux : “/usr/share/pos/config” directory
3. Reboot the terminal, the driver will update the configuration automatically. It is possible update the configuration without rebooting by running the “Update Device Firmware Tool” manually. For example

   ```
   java com.ibm.jpos.tools.sdicc.flashupdate.FlashAllDevices
   ```

   When the version number from the file is higher than the one stored at the device the new configuration will be applied.
4. Verify that the Device has the new version of configuration by using the POS Control Center and accessing the System Management tab.

Keyboard definition file

The keyboard definition file is used in conjunction with the KeyTranslationFile property. The file allows the user to map physical keys into virtual key codes. It also allows the user to configure double keys.

Virtual key codes

Each line of the table consists of an integer virtual keyboard code followed by white space (any combination of spaces and tabs) followed by the input sequence consisting of any number of actual keyboard codes. Individual codes in the input sequence are separated by white spaces.

For example, the line for up-arrow might be coded like this:

```
301 01 26 30
```

This coding indicates that when the sequence of values 01, 26, 30 comes in, recognize it as virtual keyboard code 301. (The characters corresponding to 01, 26, and 30 are **Esc**, **[**, and **A**.)

It is also possible to generate several virtual scan codes at once. This is achieved by grouping all the virtual scan codes to be generated within square brackets. The following example shows an entry indicating the drivers to generate two virtual scan codes: 302, followed by 303 when the value 141 is scanned.

```
[302 303] 141
```

The actual keyboard codes needed to generate multiple virtual scan codes can be a sequence, as in the first example:

```
[312 320 322] 141 29
```

In this case the virtual scan code sequence 312-320-322 is generated when the scan code sequence is complete. This means that the single 141 code does not trigger any scan code.

The character-encoding rules are:

1. Any single character, except space, tab, and # stands for itself. The example sequence can be coded as:

   ```
   301 01 [ A  2
   ```

2. A sequence of two or more digits is interpreted as an integer value, as in the example. If an integer starts with a zero, it is interpreted as an octal number.

   The example above can be coded as:

   ```
   301 01 032 036
   ```

   If an integer starts with a 0x, it is interpreted as a hex number.
3. An integer (decimal, hex, or octal) can have a leading plus sign.

   This is essential for coding decimal integers less than ten, because a single digit alone is interpreted as an ASCII character under Rule 1. A # character could be coded as +4 but not as a 4 only.
4. A # starts a comment. It and the rest of the line are skipped.

   If a line consists of nothing but a comment, or is empty, it is skipped.
5. The characters space, tab, and # cannot be coded literally. They can be coded as integers 57, 15, and +4 (Windows scan codes).
6. Input characters that are their own virtual keyboard codes (such as carriage return, A, Z, =, ...) do not need to be coded in the table. When the Java

Keyboards Mapping function is reading characters and is about to begin a new sequence, a character that does not match any sequence is returned as is.

7. When the finite-state machine is in the middle of some candidate sequences, a non matching character signals the end of sequence. All previous characters from beginning of sequence to the non matching character are returned as is.

   The finite-state machine then resets to accept a new sequence. For example, assume that 99 is the scan code for a, and 100 is the scan code for b. The entries in the keys.dat file are:

   ```
   400 99 b
   ```

   If you press the **a** key, and then press the **b** key, two data events are generated:

   - Press the **a** key (scan code 99). The a key does not return scan codes until scan code completes the sequence.
   - Press the **b** key. The b key does return 2 data events: scancode 99 (for a) and scancode 100 (for b); the rule is not reached.

   –

8. Virtual keycode value range from 0000 to 0xFFFF and keyboard code value range from 00 to 0xFF. If any of the values exceed its range, the line is skipped.

When making the keyboard definition table, only scan codes may be used for the system-attached keyboard. Characters cannot stand for themselves, as stated in Rule 1.

Double key support

The double key definition format is as follows:

```
0xFF <expected scan code> <scancode for 1st key> <scancode for 2nd key>
<filter time in ms>
```

The `<filter time in ms>` is optional.

Its default value is zero, and the recommended filter time is 60 milliseconds

For example:

```
0xFF 0X1C 0x71 0x1C 50
0xFF 0x7E 0x7E 0x5F 0x32
```

where:

```
0x1C is the scancode for key switch 109
0x71 is the scancode for key switch 108
0x7E is the scancode for key switch 106
0x5F is the scancode for key switch 107
```

Whenever keys 108 and 109 are pressed within 50 milliseconds of each other, the scancode returned to the application is 0x1C.

Whenever keys 106 and 107 are pressed within 50 milliseconds of each other, the scan code returned to the application is 0x7E.

The filter time specified for both double key definitions in the example above is 50 milliseconds

USB System attached keyboard on Linux

When there is a USB System keyboard, you can not have a PS/2 keyboard plugged into the system. It might cause the USB Keyboard to malfunction.

OPOS configuration

Table 127. Settings for POS keyboard

Keyword	Type	Description	OPOS Gateway Support
Numlock	String	Initial state of Num Lock light. Valid values are: **ON** Illuminated **(other value)** Dark (default) **Note:** This setting cannot be modified with the Configuration tool.	No
NumlockOn	String	Initial state of Num Lock light. Overrides the Numlock setting. Valid values are: **True** Illuminated **(other value)** Dark (default)	Yes
NumlockEnabled	String	Whether the Num Lock key is enabled. Valid values are: **True** Enabled **(other value)** Disabled (default)	Yes
ScrolllockOn	String	Initial state of Scroll Lock light. Valid values are: **True** Illuminated **(other value)** Dark (default)	Yes
ScrolllockEnabled	String	Whether the Scroll Lock key is enabled. Valid values are: **True** Enabled **(other value)** Disabled (default)	Yes

Note: The settings above apply to the physical keyboard. Since one physical keyboard can contain multiple logical devices (MSR, keylock, Line display, POS keyboard), these settings are not stored with the normal service object settings, but are stored in a different location in the registry (HKLM\SOFTWARE\OLEforRetail\ServiceInfo).

Table 128. Service Object settings for POS keyboard

Keyword	Type	Description	OPOS Gateway Support
MapPosKeys	String	Filename of Key Map file. The default value is C:\POS\BIN\KBDKMAP.DAT.	Yes
MapKeyboardToOS	String	Whether key events from RS-485 keyboard are converted to Windows key events. Valid values are: **True** Enable conversion **(other value)** Disable conversion **Note:** This setting cannot be modified with the Configuration tool.	No
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.	No

OPOS DirectIO calls

Get keyboard light status

Returns the current state of the keyboard LEDs

Table 129. Get keyboard light status

Parameter	Type	Description
Command	Int32	200
Data	Long*	Pointer to a byte. On return the byte contains a mask of the current keyboard LED state. See Table 130 for more details.
Obj	BSTR*	Ignored

Table 130. Byte information

Bit	Value	LED
0	1	Wait LED
1	2	Offline LED
2	4	Message Pending LED
3	8	Unlabeled LED

Set keyboard light on

This turns on the POS LEDs on the keyboard. This does not affect the 3 standard keyboard LEDs (Caps Lock, Scroll Lock or Num Lock).

Table 131. Set keyboard light on

Parameter	Type	Description
Command	Int32	201
Data	Long*	Pointer to a byte containing a mask of LEDs to enable
Obj	BSTR*	Ignored

Set keyboard light off

This turns off the POS LEDs on the keyboard. This does not affect the 3 standard keyboard LEDs (Caps Lock, Scroll Lock or Num Lock).

Table 132. Set keyboard light off

Parameter	Type	Description
Command	Int32	202
Data	Long*	Pointer to a byte containing a mask of LEDs to disable
Obj	BSTR*	Ignored

Additional OPOS information

Configuring the Alphanumeric Point of Sale keyboard

If you chose to install your IBM Alphanumeric Point-of-Sale (ANPOS) or Compact Alphanumeric Point-of-Sale (CANPOS) keyboard as the system keyboard, you should use the ANPOS utility program to:

1. Define double keys. These key switches can be doubled: 77, 78, 82, 87, 88, 90, 94, 95, 99, 100, 105 to 109, 112 to 123, 125 to 128.
2. Override the keyboard default values for the following settings:
   - Numeric keypad zero (key 94, 99)
   - Key click
   - POS LEDs initial setting

The ANPOS utility runs automatically at boot time. It uses the sample resource file, `aipsys.res`.

- On Windows, aipsys.res is located in the default install directory, C:\POS.
- On Linux systems, aipsys.res is located in the /etc directory.

Notes:

1. In the aipsys.res file, the keyboard resource names must be qualified with the application name, **aipanpos**, and the device name, **system**. For example:

   ```
   aipanpos.system.keyboardClick: SOFT
   aipanpos.system.doubleKey01:  77,82
   aipanpos.system.doubleKey02:  90,95
   ```

   See “Modifying device behavior (USB, RS-485 and PS/2 devices)” on page 50 for more information on using the resource file.
2. The ANPOS utility records any errors in the file, aipanpos.log.
   - On Windows systems, `aipanpos.log` is created in the default log directory, C:\POS\LOG.
   - On Linux systems, `aipanpos.log` is created in the /var/log directory.
3. For a USB system keyboard, the ANPOS utility is used only for:
   - Double key definition
   - LED setting (initial)
   - Key click setting

   The POS LEDs and the key click setting are not supported for the USB System Keyboard.
4. Changes to the aipsys.res file will not take effect until the system is rebooted.

SureOne/SurePOS 100 keyboard

The keyboard on the SureOne is a standard PS/2 keyboard. IBM OPOS does not implement the UPOS POS keyboard specification for this keyboard.

Mapping POS keys

To configure the keyboard, perform the following steps:

1. Select a POS keyboard entry and click **Configure**. The keyboard configuration dialog is displayed (see Figure 22).

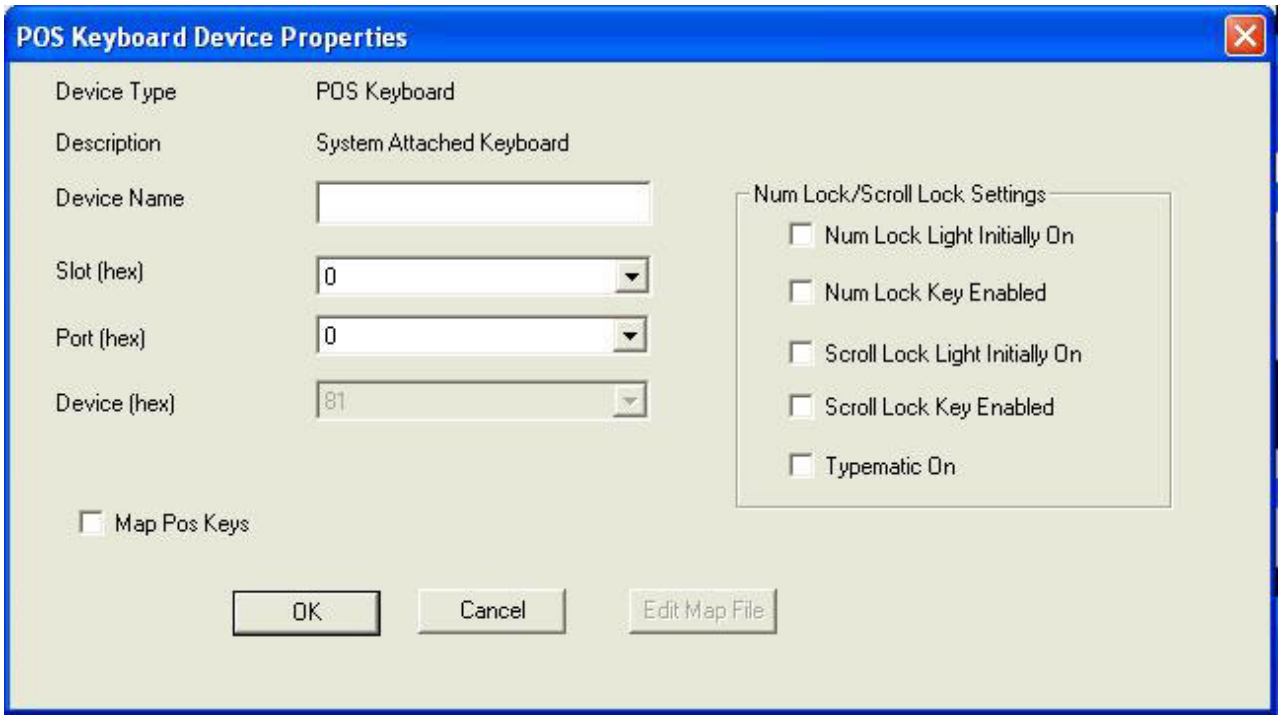


Figure 22. Keyboard properties dialog

Note: There is a single entry for System Attached Keyboard. This entry can be used for both PS/2 and USB system keyboards. For a PS/2 system keyboard, the Slot value must be 0; for a USB system keyboard, the Slot value must be 8.

2. To configure POS keyboard mapping, select **Map Pos Keys** and click the **Edit Map File** button.

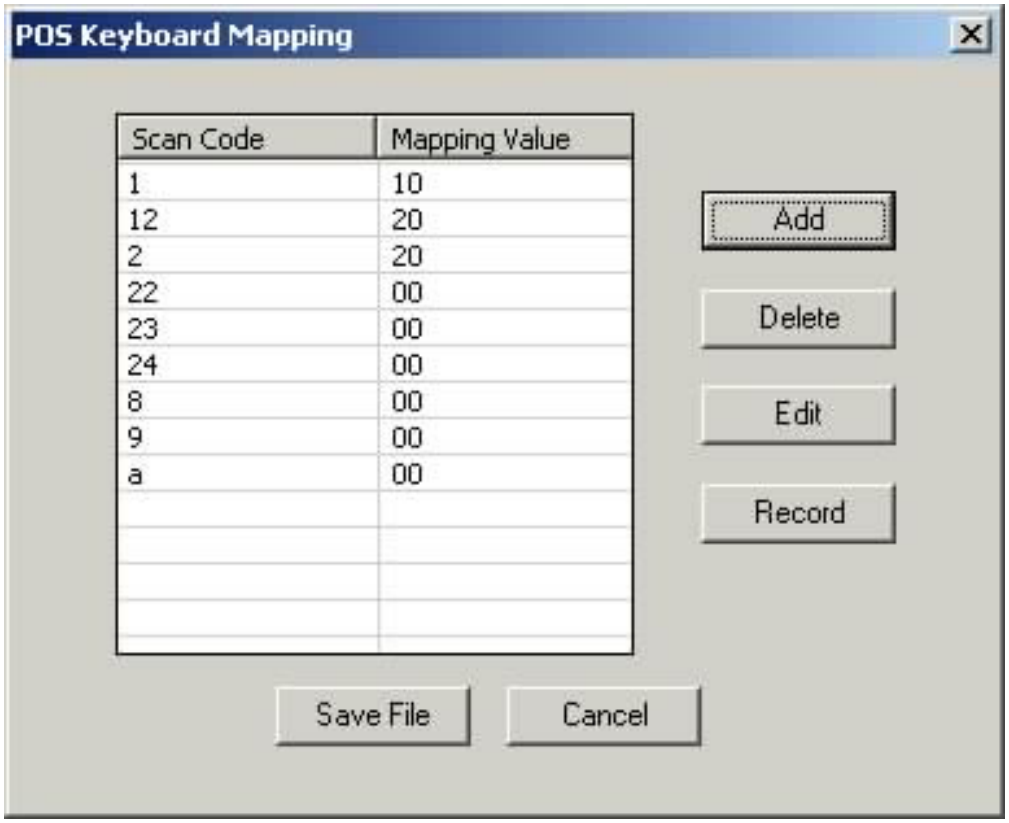


Figure 23. Sample POS keyboard mapping dialog

3. Select or clear the **Num Lock** and **Scroll Lock** to specify the initial settings you want for those keys.
4. To configure POS keyboard mapping, select **Map Pos Keys** and click the **Edit Map File** button. A filename field is displayed.
5. Enter the name of the POS keyboard mapping file. If the file exists, it is read and the values are displayed in the POS Keyboard Mapping window (see Figure 23).
6. Edit the mapping data.
   - To create an entry, click the **Add** button. A new entry is inserted at the top of the list. Enter the scan code and mapping value, then click **Save**.

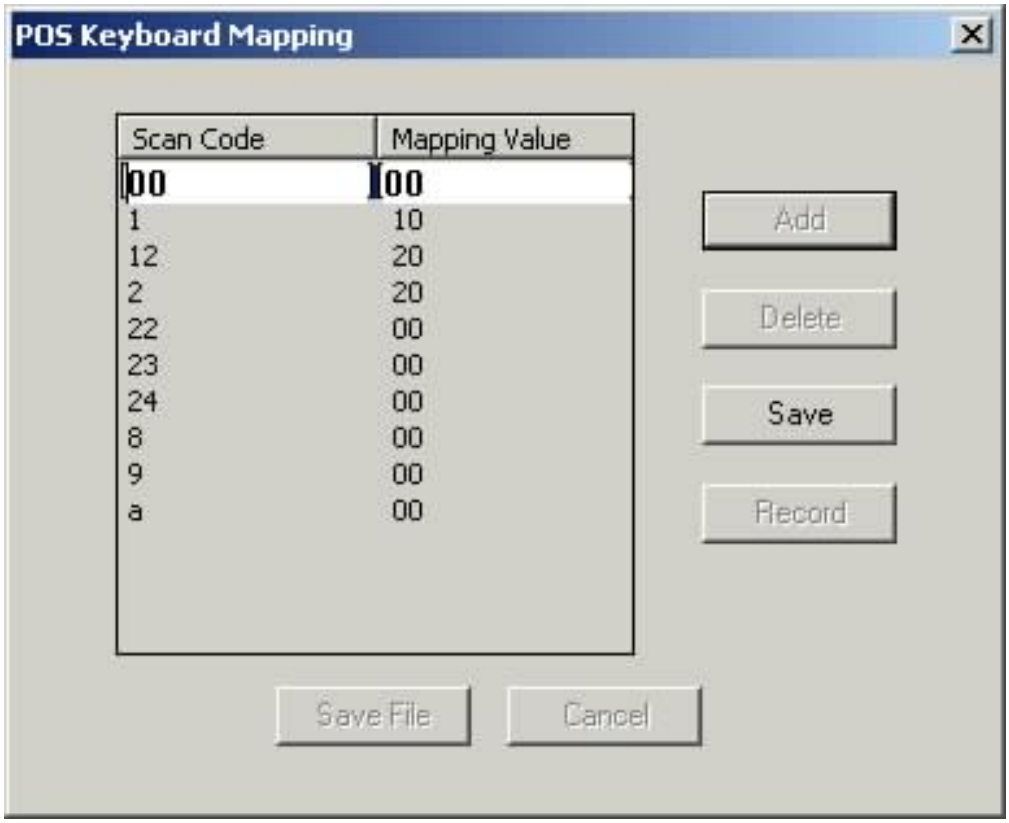


Figure 24. Example of adding a keyboard mapping entry

Note: Non-unique entries are allowed. If multiple entries with the same name exist, only the first entry is used.

- To modify an existing entry, select it and click the **Edit** button. Edit the values and click **Save**.

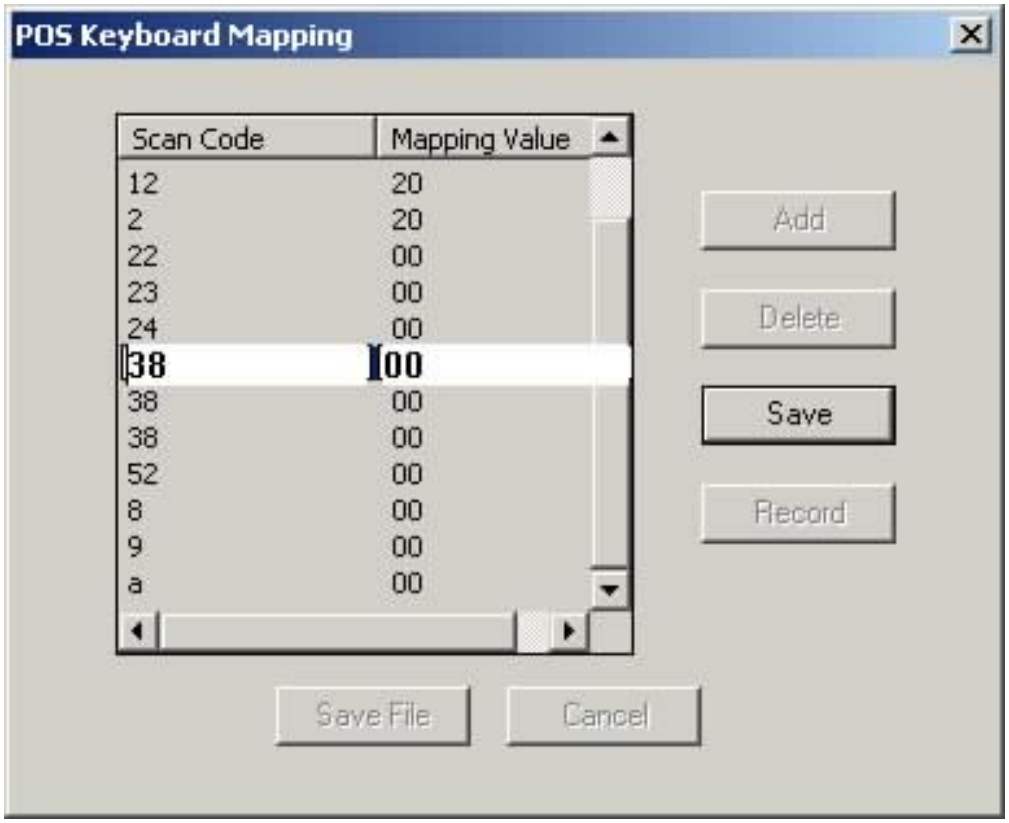


Figure 25. Editing a keyboard mapping entry

- To remove an entry, select it and click **Delete**.
- To record the scan code of a key, click **Record** and press the key you want to record. An entry is inserted in the list with the scan code of the key you pressed. The dialog continues recording keys until you click the **Stop** button. Use the **Edit** menu to assign mapping values to the new entries.

7. When you have finished editing the keyboard mapping information, click **Save File**. The keyboard mapping file is saved.

Chapter 16. POS printer

Supported devices

Table 133. POS printer supported devices

Device	Connectivity	Comments
1. 4610 printer family	EIA-232	Includes Tx1, Tx2, Tx3, Tx4, Tx5, Tx6, Tx7, Tx8, Tx9 models
2. 4610 printer family	RS-485	Includes Tx1, Tx2, Tx3, Tx4, Tx5, Tx6, Tx7, Tx8, Tx9, models
3. 4610 printer family	USB	Includes Tx1, Tx2, Tx3, Tx4, Tx5, Tx6, Tx7, Tx8, Tx9, models
4. Model 3/4 printer family	RS-485	Includes 4, 4R, 4A
5. Model 3/4 printer family	USB	Printer Protocol Converter is used, include 4, 4R, 4A
6. 4689 printer family (impact)	RS-485	Includes 001, 002
7. 4689 printer family (thermal)	RS-485	Includes 301, 3G1, 3M1 (OPOS only)
8. 4689 printer family (thermal)	RS-485, USB	TD5
9. 4674 built-in printer	RS-485	TD5 integrated
10. SureOne printer (single-head impact)	EIA-232	OPOS only
11. SureOne and SurePOS 100 printer (thermal)	EIA-232	
12. SureOne printer (double-head impact)	EIA-232	OPOS only
13. SureOne printer (A04/A05 impact) and SurePOS 100 printer (impact)	EIA-232	
14. 4610 2xR/1NR	EIA-232	
15. 4610 2xR/1NR	RS-485	
16. 4610 2xR/1NR	USB	

Supported properties and methods

Table 134. POS printer common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	1 true only for SST/TI3/TI4/TI5/TI8/TI9/2NR/2CR/1NR
CapPowerReporting	PR_STANDARD	PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	1 true only for SST/TI3/TI4/TI5/TI8/TI9/2NR/2CR/1NR
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False

Table 134. POS printer common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CheckHealthText	All	
Claimed	All	
DataCount	Not supported	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	All	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 135. POS printer specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	
CapCharacterSet	All - Values based on HW	
CapConcurrentJrnRec	4, 5, 6, 7, 8, 9	4, 5, 6, 7, 8
CapConcurrentJrnSlp	Not supported	
CapConcurrentPageMode	Not supported	
CapConcurrentRecSlp	1, 2, 3 (TI1-5, TI8-9), 14, 15, 16 (2xR)	1, 2, 3 (TI1-5, TI8–9), 14,15,16 (2xR)
CapCoverSensor	All	All except SureOne printers
CapMapCharacterSet	Not supported	1, 2, 3 (TI1-5, TI8–9), 14,15,16 (2xR, 1NR)
CapRecPageMode	1, 2, 3, 14, 15, 16	1, 2, 3 true only for TI4/TI8/TI9, TI5/TI9, 2CR, 2NR, 1NR
CapSlpPageMode	Not supported	
CapTransaction	All	
CapJrnPresent	4, 5, 6, 7, 8, 9	4, 5, 6, 7, 8
CapJrn2Color	Not supported	
CapJrnBold	4, 5, 7, 8, 9	4, 5
CapJrnCartridgeSensor	Not supported	
CapJrnColor	Not supported	

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapJrnDhigh	4, 5, 7, 8, 9	4, 5, 7, 8
CapJrnDwide	4, 5, 7, 8, 9	4, 5, 7, 8
CapJrnDwideDhigh	4, 5, 7, 8, 9	4, 5, 7, 8
CapJrnEmptySensor	4, 5, 6, 7, 8, 9	4, 5, 6, 7, 8
CapJrnItalic	Not supported	
CapJrnNearEndSensor	7, 8, 9	7, 8
CapJrnUnderline	7, 8, 9	7, 8
CapRecPresent	All	All
CapRec2Color	1, 2, 3 (TI3-4,Tx6-9), 14, 15, 16 EC > 0*x*33	
CapRecBarCode	1, 2, 3, 7, 8, 9, 11, 14, 15, 16	1, 2, 3, 7, 8, 11, 14, 15, 16
CapRecBitmap	All	
CapRecBold	1, 2, 3, 4, 5, 7, 8, 9, 11, 13, 14, 15, 16	1, 2, 3, 4, 5, 10, 11, 13, 14, 15, 16
CapRecCartridgeSensor	Not supported	
CapRecColor	1, 2, 3 (TI3-4,Tx6-9), 14, 15, 16 EC > 0x33	
CapRecDhigh	1, 2, 3, 7, 8, 9, 11, 13, 14, 15, 16 4, 5 DH forces DW reverts to normal Rotation	1, 2, 3, 7, 8, 11, 13, 14, 15, 16 4, 5 DH forces DW 10 reverts to normal rotation in 180 mode
CapRecDwide	1, 2, 3, 4, 5, 7, 8, 9, 11, 13, 14, 15, 16	1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16
CapRecDwideDhigh	1, 2, 3, 4, 5, 7, 8, 9, 11, 13, 14, 15, 16	1, 2, 3, 4, 5, 7, 8, 11, 13, 14, 15, 16 10 reverts to normal rotation in 180 mode
CapRecEmptySensor	1, 2, 3, 6, 7, 8, 9, 14, 15, 16	1,2,3 (2CR, 2NR, 1NR models only) 6, 14, 15, 16
CapRecItalic	Not supported	
CapRecLeft90	1, 2, 3, 14, 15, 16	
CapRecMarkFeed	Not supported	
CapRecNearEndSensor	7, 8, 14, 15, 16 (2xR)	
CapRecPapercut	All	All except 10, 12, and 13
CapRecRight90	1, 2, 3, 8, 9, 14, 15, 16	1, 2, 3, 8, 14, 15, 16
CapRecRotate180	1, 2, 3, 11, 14, 15, 16 EC > 0*x*33	1, 2, 3 EC > 0*x*33 and 10, 11, 12, 13, 14, 15, 16
CapRecStamp	1, 2, 3, 11, 13, 14, 15, 16 - uses stored bitmap 1 6 - physical stamp 7, 8, 9 - downloaded stamp	
CapRecUnderline	1, 2, 3, 7, 8, 9, 11, 13, 14, 15, 16	1, 2, 3, 7, 8, 10, 11, 12, 13, 14, 15, 16
CapSlpPresent	1, 2, 3 (TI1-5,TI8-9), 4, 5, 14, 15, 16 (2xR)	1, 2, 3, 4, 5, 14, 15, 16 (2xR)
CapSlpFullslip	4, 5	
CapSlp2Color	Not supported	

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapSlpBarCode	1, 2, 3, 14, 15, 16 (2xR) EC > 0*x*1D	
CapSlpBitmap	1, 2, 3 (TI1-5,TI8-9), 4, 5, 14, 15, 16 (2xR)	1, 2, 3, 4, 5, 6, 14, 15, 16 (2xR)
CapSlpBold	1, 2, 3 (TI1-5,TI8-9), 4, 5, 14, 15, 16 (2xR)	1, 2, 3, 4, 5, 14, 15, 16 (2xR)
CapSlpBothSidesPrint	1, 2, 3, (TI1-5,TI8-9), 14, 15, 16 (2CR)	
CapSlpCartridgeSensor	Not supported	
CapSlpColor	Not supported	
CapSlpDhigh	1, 2, 3 4, 5 DH forces DW 14, 15, 16 (2xR)	
CapSlpDwide	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	
CapSlpDwideDhigh	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	
CapSlpEmptySensor	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	
CapSlpItalic	Not supported	
CapSlpLeft90	1, 2, 3, 14, 15, 16 (2xR)	1, 2, 3 (except TI5 and Tx7), 14,15,16 (2xR - SBCS Only)
CapSlpNearEndSensor	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	1, 2, 3, 4, 5, 6, 14, 15, 16 (2xR)
CapSlpRight90	Not supported	
CapSlpRotate180	Not supported	
CapSlpUnderline	Not supported	
CartridgeNotify	Not supported	
CharacterSet	All	

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CharacterSetList	1, 2, 3 - "437, 819, 850, 852, 855, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 998, 999, 1116, 1117, 1118, 1119, 1250, 1251, 1252, 1253, 1254, 1257" 8, 9 - "437, 858, 998" 11 - All models support this list - ″437, 850, 852, 855, 857, 862, 864, 866, 874, 897, 998″ DBCS models have additional character sets: • Japanese: 932 • Korean: 1361 • traditional Chinese: 950 • simplified Chinese: 1381 13 – "437, 850, 852, 855, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 874, 897, 998" for Standard model. 14,15,16 - ″437, 775, 819, 848, 850, 852, 855, 857, 858,860, 861, 862, 863, 864, 865,866, 867, 869, 872, 998, 999, 1116, 1117, 1118, 1119, 1250, 1251, 1252, 1253, 1254, 1255, 1256,1257" DBCS models have "437, 998" and one additional DBCS character set: • Japanese: 932 • Korean: 1361 • traditional Chinese/Taiwanese: 950 • simplified Chinese: 1381	1 (TI1-8) - ″101, 437, 850, 858, 860, 863, 865, 998″ 1 (2CR,2NR,1NR) -″101, 437, 775, 850, 858, 860, 863, 865, 869, 857, 867, 852, 848, 866, 872, 861, 998, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257″ 2, 3 (TI1-8)- ″437, 819, 850, 852, 855, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 998, 999, 1116, 1117, 1118, 1119, 1250, 1251, 1252, 1253, 1254, 1257″ 2, 3 (2CR,2NR,1CR)- ″437, 775, 819, 848, 850, 852, 855, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 872, 998, 999, 1116, 1117, 1118, 1119, 1250, 1251, 1252, 1253, 1254, 1256, 1257″ 8, 9 - ″437, 858, 998″ 11 - All models support this list - ″437, 850, 852, 855, 857, 862, 864, 866, 874, 897, 998″ DBCS models have additional character sets: • Japanese: 932 • Korean: 1361 • traditional Chinese: 950 • simplified Chinese: 1381 13 - ″437, 850, 852, 855, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 874, 897, 998″ for Standard model. DBCS models have ″437, 998″ and one additional DBCS character set: Japanese: 932 Korean: 1361 traditional Chinese: 950 simplified Chinese: 1381
CoverOpen	1, 2, 3, 4, 5, 11, 14, 15, 16 7, 8, 9 - when printer not idle	1, 2, 3, 4, 5, 14, 15, 16 7, 8 - when printer not idle
ErrorLevel	All	
ErrorStation	All	
ErrorString	All	
FlagWhenIdle	All	
FontTypefaceList	Not supported	1, 2, 3 (USB Windows) (TI3-5, Tx6-9) EC > 0*x*33 can list ″Fixed, Proportional″ based on downloaded UD Fonts, null string otherwise
JrnCartridgeState	Not supported	
JrnCurrentCartridge	Not supported	
JrnEmpty	4, 5, 6, 7, 8. 9	4,5,6,7,8
JrnLetterQuality	Not supported	

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
JrnLineChars	1, 2, 3, 14, 15, 16 - 0 4, 5 - 38 6 - 25 7, 8, 9 - 32	1, 2, 3, 10, 11, 12, 13, 14, 15, 16 - 0 4, 5 - 38 6 - 25 7, 8 - 32
JrnLineCharsList	1, 2, 3, 14, 15, 16 - ″″ (empty) 4, 5 - ″30,38″ 6 - ″25,30″ 7, - ″32,42″ 8, 9 - "32"	1, 2, 3, 10, 11, 12, 13, 14, 15, 16 - ″″ (empty) 4, 5 - ″30,38″ 6 - ″25,30″ 7, 8- ″32,42″
JrnLineHeight	1, 2, 3, 14, 15, 16 - 0 4, 5, 6 - 9 7, 8, 9 - 24	1, 2, 3, 10, 11, 12, 13, 14, 15, 16 - 0 4, 5, 6 - 9 7, 8- 24
JrnLineSpacing	1, 2, 3, 14, 15, 16 - 0 4, 5, 6 - 12 7, - 27 8, 9 - 30	1, 2, 3, 10, 11, 12, 13 - 0 4, 5, 6 - 12 7, 8- 27
JrnLineWidth	1, 2, 3, 14, 15, 16 - 0 4, 5 - 380 6 - 300 7 - 420 8, 9 - 432	1, 2, 3, 10, 11, 12, 13, 14, 15, 16 - 0 4, 5- 380 6 - 300 7, 8 - 420
JrnNearEnd	7, 8, 9	7, 8
MapCharacterSet	Not supported	1, 2, 3, 14, 15, 16
MapMode	All	
PageModeArea	4610: When PageModeStation is set ″0,0,576,1250″ (for TI8/9/2xR/1NR) or ″0,0,576,800″; else zero	1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - When PageModeStation is set ″0,0,576,1250″ (for TI8/9) or ″0,0,576,800″; else zero

PageModeDescriptor	4610:
	`PTR_PM_BITMAP |`
	`PTR_PM_BARCODE |`
	`PTR_PM_BM_ROTATE |`
	`PTR_PM_BC_ROTATE |`
	`PTR_PM_OPAQUE`
	
	When PageModeStation is set; else zero.
		1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR)
		`PTR_PM_BITMAP |`
		`PTR_PM_BARCODE |`
		`PTR_PM_BM_ROTATE |`
		`PTR_PM_BC_ROTATE |`
		`PTR_PM_OPAQUE`
		
		When PageModeStation is set; else zero.

PageModeHorizontalPosition	4610: Zero before setting PageModeStation	1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - Zero before setting PageModeStation

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
PageModePrintArea	4610: When PageModeStation is set, max is PageArea value, else zero	OPOS: 1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - When PageModeStation is set, max is PageArea value, else zero
PageModePrintDirection	4610: Initialized to jpos.POSPrinterConst.PTR_PD_LEFT_TO_RIGHT when the device is first enabled	1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - Initialized to PTR_PD_LEFT_TO_RIGHT when the device is first enabled
PageModeStation	4610: jpos.POSPrinterConst.PTR_S_RECEIPT	1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - PTR_S_RECEIPT
PageModeVerticalPosition	4610: Zero before setting PageModeStation	1,2,3 (TI4/TI8/TI9, TI5/TI9 EC > 0x33, 2CR, 2NR, 1NR) - Zero before setting PageModeStation
RecBarCodeRotationList	1, 2, 3, 14, 15, 16 (TI3-5, Tx6-9) EC > 0*x*33 can list ″0,180″. Otherwise, 1, 2, 3 list ″0″. 8, 9 - "0" 14, 15, 16 "0, 180"	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 can list ″0,180″. Otherwise, 1, 2, 3 list ″0″.
RecBitmapRotationList	″0″	
RecCartridgeState	Not supported	
RecCurrentCartridge	Not supported	
RecEmpty	1, 2, 3, 6, 7, 8, 9, 14, 15, 16	1, 2, 3, 6, 14, 15, 16
RecLetterQuality	1, 2, 3 (TI3-5, Tx6-9), 14, 15, 16 EC > 0*x*33	
RecLineChars	1, 2, 3 - (TI5,Tx7) - 38, (all other) 44 4, 5, 6, 7 - 38 11 - 48 13 – 40 with narrow paper 28 8, 9 - 32 14, 15, 16 (DBCS) - 38 14, 15, 16 (SBCS) - 44	1, 2, 3 (TI3-5, Tx6-9 EC > 0x33, 2CR, 2NR, 1NR) - 56 1, 2, 3 (TI5,Tx7) - 38 (all other) - 44 4, 5, 6, 7, 8 - 38 10, 11, 12, 13 - 40

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
RecLineCharsList	1, 2, 3 - EC > 0x33 (TI5,Tx7) - 28, 30, 32, 33, 36, 38, 41, 44, 48 (TI4, Tx6, TI8, TI9, 2xR/1NR - SBCS) - "32, 34, 36, 38, 41, 44, 48, 52, 57, 64, 72" 1, 2, 3 (All other), - "34, 44, 48" 4, 5 - "30,38" 6 - "25,30" 7 - "32,42" 8, 9, ″32″ 11- "36,38,41,48" 13 – “22,33,40” with narrow paper “15,23,28” 14,15,16 (DBCS) - "28,30,32,33,36,38,41,44,48" 14,15,16 (SBCS) - "32,34,36,38,41,44,48,52,57,64,72"	1 (TI3-5, Tx6-9) EC > 0*x*33 - ″34, 44, 48, 57″ (TI3-5, Tx6-9) EC > 0x33 -> (TI3-5, Tx6-9) EC > 0x33, 2CR, 2NR, 1NR 1, 2, 3 (All other), 10 - ″34, 44, 48″ 4, 5 - ″30, 38, 42″ (older models only support ″30, 38″) 6 - ″25, 30″ 7, 8 - ″32, 42″ 10, 13 - ″22, 33, 40″ 11 - ″36, 38, 44, 48″ 12 - ″33, 40″
RecLineHeight	1, 2, 3 (TI5,Tx7) - 24, (All other) - 20 4, 5, 6 - 9 7, 8, 9 - 24 11 - 24 13 - 9 14, 15, 16 (DBCS) - 24 14, 15, 16 (SBCS) - 20	1, 2, 3, 14, 15, 16 - 20
RecLineSpacing	1, 2, 3 - 34 4, 5, 6 - 12 7 - 27 8, 9 - 30 11 - 32 13 - 11 14, 15, 16 - 34	1, 2, 3 - 34 4, 5, 6 - 12 7, 8 - 27 10, 12, 13 - 9 11 - 32 14, 15, 16 - 34

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
RecLinesToPaperCut	1, 2, 3 - 5 4, 5, 7 - 6 8, 9 - 3 6 - 16 11, 13 - 5 14, 15, 16 - 5	1 - 7 2, 3 - 12 4, 5, 7, 8 - 6 6 - 16 10, 11, 12, 13 - 4
RecLineWidth	1, 2, 3 (TI1-5, Tx6-9) - 576 1, 2, 3 Tx6-9 Narrow Paper - 400 4, 5 - 380 6 - 300 7 - 420 8, 9 - 320 11 - 576 13 – 200, with narrow paper 140 14, 15, 16 – 576 14, 15, 16 (1NR) narrow paper- 400	1, 2, 3 (TI1-5, Tx6-9) - (TI1-5, Tx6-9 2CR, 2NR, 1NR) 1, 2, 3 Tx6-9 narrow paper - 400 4, 5 - 380 6 - 300 7, 8 - 420 10, 12, 13 - 280 11 - 576
RecNearEnd	1, 2, 3, 7, 8, 14, 15, 16	14, 15, 16
RecSidewaysMaxLines	1, 2, 3 TI1-5, Tx6-9 - 17 1, 2, 3 Tx6-9 Narrow Paper - 12 7, 8, 9 - 14 14, 15, 16 – 17 14, 15, 16 (1NR) narrow paper- 12	1, 2, 3 TI1-2-25, T13-5,Tx9 -17 1, 2, 3 Tx6-7 Narrow Paper - 11 7, 8 - 19 14, 15, 16 - 17
RecSidewaysMaxChars	1, 2, 3 - 61, Ti8/9 – 95 7, 8, 9 – 256 14, 15, 16 – 95	1, 2, 3 TI1-2 - 256, TI3-5 - 53, Tx6-Tx7 - 61, Tx8-Tx9 - 96 7, 8 - 53 14, 15, 16 - 96
RotateSpecial	_NORMAL: 1, 2, 3, 8, 9, 11, 14, 15, 16 _LEFT90: — _RIGHT90: — 8, 9 _ROTATE180: 1, 2, 3 (TI3-5, Tx6-9) EC > 0x33, 14, 15, 16	OPOS: _NORMAL: 1, 2, 3 _LEFT90: -- _RIGHT90: -- _ROTATE180: 1, 2, 3 (TI3-5, Tx6-9) EC > 0x33, 2CR, 2NR, 1NR
SlpBarCodeRotationList	″0″	1, 2, 3 - (TI3-5, Tx6-9) EC > 1D, 2CR, 2NR, 1NR, has ″0″
SlpBitmapRotationList	″0″	

Table 135. POS printer specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
SlpCartridgeState	Not supported	
SlpCurrentCartridge	Not supported	
SlpEmpty	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	
SlpLetterQuality	1, 2, 3 (TI3-5, Tx6-9), EC > 0x33 14, 15, 16	1, 2, 3 (TI3-4, Tx6-9, TI8, TI9) EC > 0x33 (not supported on TI5) 2CR, 2NR
SlpLineChars	1, 2, 3 - (TI5)- 24, (all other) 47 4, 5 - 86 14, 15, 16 (2xR-DBCS) - 24 14, 15, 16 (2xR-SBCS) - 47	1, 2, 3- (TI5) -24 (all other) - 47 4, 5 - 86 6 - 58
SlpLineCharsList	1, 2, 3 - (TI5) - ″19, 23, 24, 26, 27, 29″, (all other) ″30, 37, 38, 42, 47, 52″ 4, 5 - ″30, 38, 42, 68, 86, 94″ — older models only support 30, 38, 68, 86 14, 15, 16 (2xR-DBCS) - "19, 23, 24, 26, 27, 29" 14, 15, 16 (2xR-SBCS) - "30, 37, 38, 42, 47, 52"	1 - (TI5,Tx7) - ″19, 24″ (all other)"37, 47, 52" 2, 3 - (TI5, Tx7) - ″23, 24, 26, 27, 29″ (all other)"30, 37, 38, 42, 47, 52" 4, 5 - ″30, 38, 42, 68, 86, 94″ — older models support only 30, 38, 68, 86 6 - ″58, 70″
SlpLineHeight	1, 2, 3, 14, 15, 16 (2xR) - 7 4, 5 - 9	1 - 4 2, 3 - 7 4, 5, 6 - 9
SlpLinesNearEndToEnd	All - values based on HW	
SlpLineSpacing	1, 2, 3, 14, 15, 16 (2xR) - 9 4, 5 - 12	1 - 9 2, 3 - 8 4, 5, 6 - 12
SlpLineWidth	1, 2, 3, 14, 15, 16 (2xR) - 474 4, 5 - 880	1 - 474 2, 3 - 470 4, 5 - 880 6 - 300
SlpMaxLines	All - values based on HW	
SlpNearEnd	1, 2, 3, 4, 5, 14, 15, 16 (2xR)	1, 2, 3, 4, 5, 6
SlpPrintSide	1, 2, 3 (TI1–5, TI8–9), 14, 15, 16 (2xR)	
SlpSidewaysMaxLines	1, 2, 3, 14, 15, 16 (2xR) - 19	1 - 18 2, 3 - 19
SlpSidewaysMaxChars	1, 2, 3, 14, 15, 16 (2xR) - 148	1, 2, 3 - 147

Table 136. POS printer common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	INTERNAL = ALL EXTERNAL = ALL INTERACTIVE = Not supported	All
claim	All	
clearInput	Not supported	
clearOutput	All	
close	All	
compareFirmwareVersion	Not supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9/2CR/2NR/1NR
directIO	For 1, 2, 3 only: **X'01'** Flip check 8, 9 - **X'02'** PRINT_DOWNLOADED_BITMAP_ID **X'09'** DOWNLOAD_NON_PROP_FONT_ID **X'23'** ERASE_FLASH_MEMORY_SECTOR_ID **X'20'** SET_STAMP_ID **X'25'** DOWNLOAD_DBCS_FONT_ID	For 1, 2, 3 only: **X'01'** Flip check **X'10'** Write flash memory **X'11'** Read flash memory **X'12'** Query flash size **X'13'** Query maximum records **X'14'** Set record length **X'15'** Erase flash memory **X'16'** Get record length **101** Clear page mode print area **102** Set/get page mode print **103** Set/get horizontal print position (page mode) **104** Set/get vertical print position (page mode) **105** Get page area **106** Set/get page mode station **107** Set/get page mode print area **108** Set/get print direction **201** Set/get MICR read with scan **202** Set/get compression format **203** Scan document **204** Store document **205** Retrieve by index **206** Retrieve by tagname **207** Erase image storage
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Supported: 4610 and SureOne
updateFirmware	Not supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9/2NR/2CR
updateStatistics	Not supported	

Table 137. POS printer specific methods

Method	JavaPOS and OPOS Gateway	OPOS
beginInsertion	1, 2, 3 (except SST), 4, 5	1, 2, 3, 4, 5, 6
beginRemoval	1, 2, 3 (except SST), 4, 5	1, 2, 3, 4, 5, 6
changePrintSide	1, 2, 3 except SST and TI5	1, 2, 3 except SST and TI5, 1NR
clearPrintArea	4610: supported	Not supported
cutPaper	1, 2, 3, 7, 8, 9	All except 11
endInsertion	1, 2, 3 (except SST), 4, 5	1, 2, 3, 4, 5, 6
endRemoval	1, 2, 3 (except SST), 4, 5	1, 2, 3, 4, 5, 6
markFeed	Not supported	
pageModePrint	4610: Control is one of the following: PTR_PM_PAGE_MODE, PTR_PM_PRINT_SAVE, PTR_PM_NORMAL, PTR_PM_CANCEL	
printBarCode	1, 2, 3, 7, 8, 9, 11 (receipt) 1, 2, 3 (TI3-5, Tx6-9) EC>1D (slip)	1, 2, 3, 11 (receipt) 1, 2, 3 (TI3-5, Tx6-9) EC>1D (slip)
printBitmap	All 8, 9 (receipt)	All
printImmediate	All	
printMemoryBitmap	All	
printNormal	All	
printTwoNormal	All except 11, 13	1, 2, 3, 4, 5, 6, 7, 8, 9
rotatePrint	_NORMAL: All _LEFT90: 1, 2, 3 _RIGHT90: 1, 2, 3 (receipt), 7, 8, 9 (receipt) _ROTATE180: 1, 2, 3, (TI3-5, Tx6-9) EC > 0*x*33 (receipt), 11, 13	_NORMAL: All _LEFT90: 1, 2, 3 (slip) _RIGHT90: 1, 2, 3, 7, 8 (receipt) _ROTATE180: 1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 2CR, 2NR, 1NR(receipt), 10, 11, 12, 13 (receipt)
setBitmap	1, 2, 3, 4, 5, 7, 8, 9, 11, 13	All
setLogo	All	
transactionPrint	All	
validateData	All	

Table 138. POS printer events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All	1, 2, 3
ErrorEvent	All	
OutputCompleteEvent	All	
StatusUpdateEvent:		
PTR_SUE_COVER_OK	All	All
PTR_SUE_COVER_OPEN	All	All
PTR_SUE_JRN_COVER_OK	4, 5	Not supported
PTR_SUE_JRN_COVER_OPEN	4, 5	Not supported
PTR_SUE_REC_COVER_OK	1, 2, 3, 4, 5, 11, 14, 15, 16	Not supported
PTR_SUE_REC_COVER_OPEN	1, 2, 3, 4, 5, 11, 14, 15, 16	Not supported
PTR_SUE_SLP_COVER_OK	1, 2, 3 except SST 14, 15, 16 except 1NR	Not supported
PTR_SUE_SLP_COVER_OPEN	1, 2, 3 except SST 14, 15, 16 except 1NR	Not supported
PTR_SUE_JRN_EMPTY	4, 5, 8, 9	4, 5, 8, 9
PTR_SUE_JRN_NEAREMPTY	8, 9	8, 9
PTR_SUE_JRN_PAPEROK	4, 5, 8, 9	4, 5, 8, 9
PTR_SUE_REC_EMPTY	8, 9, 14, 15, 16	1, 2, 3(Only 2CR-2NR), 8, 9
PTR_SUE_REC_NEAREMPTY	8, 9, 14, 15, 16	1, 2, 3(Only 2CR-2NR), 8, 9
PTR_SUE_REC_PAPEROK	8, 9, 14, 15, 16	1, 2, 3(Only 2CR-2NR), 8, 9
PTR_SUE_SLP_EMPTY	1, 2, 3, 4, 5 except SST 14, 15, 16 except 1NR	1, 2, 3, 4, 5
PTR_SUE_SLP_NEAREMPTY	1, 2, 3, 4, 5 except SST 14, 15, 16 except 1NR	1, 2, 3, 4, 5
PTR_SUE_SLP_PAPEROK	1, 2, 3, 4, 5 except SST 14, 15, 16 except 1NR	1, 2, 3, 4, 5
PTR_SUE_IDLE	All	All
PTR_SUE_JRN_CARTRIDGE_EMPTY	Not Supported	Not Supported
PTR_SUE_JRN_HEAD_CLEANING	Not Supported	Not Supported
PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	Not Supported	Not Supported
PTR_SUE_JRN_CARTRIDGE_OK	Not Supported	Not Supported
PTR_SUE_REC_CARTRIDGE_EMPTY	Not Supported	Not Supported
PTR_SUE_REC_HEAD_CLEANING	Not Supported	Not Supported
PTR_SUE_REC_CARTRIDGE_NEAREMPTY	Not Supported	Not Supported
PTR_SUE_REC_CARTRIDGE_OK	Not Supported	Not Supported

Table 138. POS printer events (continued)

Event	JavaPOS and OPOS Gateway	OPOS
PTR_SUE_SLP_CARTRIDGE_EMPTY	Not Supported	Not Supported
PTR_SUE_SLP_HEAD_CLEANING	Not Supported	Not Supported
PTR_SUE_SLP_CARTRIDGE_NEAREMPTY	Not Supported	Not Supported
PTR_SUE_SLP_CARTRIDGE_OK	Not Supported	Not Supported
OPOS_CFV_FIRMWARE_OLDER	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_CFV_FIRMWARE_SAME	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_CFV_FIRMWARE_NEWER	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_CFV_FIRMWARE_DIFFERENT	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_CFV_FIRMWARE_UNKNOWN	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_PROGRESS	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_COMPLETE	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_FAILED_DEV_OK	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
OPOS_SUE_UF_FAILED_DEV_UNKNOWN	Not Supported	1 true only for SST/TI3/TI4/TI5/TI8/TI9
IBM_JPOS_SUE_PTR_REC_UNEXPECTED_COVER_OPEN	14,15,16	All
IBM_JPOS_SUE_PTR_SLP_UNEXPECTED_COVER_OPEN	14,15,16	All
IBM_JPOS_SUE_PTR_MAIN_LOGIC_CARD_FAILURE	14,15,16	Not Supported
IBM_JPOS_SUE_PTR_INTERFACE_LOGIC_CARD_FAILURE	14,15,16	Not Supported
IBM_JPOS_SUE_PTR_THERMALREC_PRINT_HEAD_FAILURE	14,15,16	Not Supported
IBM_JPOS_SUE_PTR_IMPACTSLP_PRINT_HEAD_FAILURE	14,15,16	Not Supported
IBM_JPOS_SUE_PTR_PAPER_MOTION_SENSOR_FAILURE	14,15,16	Not Supported

Table 139. POS printer BarCode symbologies supported

BarCode Symbology	JavaPOS and OPOS Gateway	OPOS
PTR_BCS_UPCA	1, 2, 3, 8, 9, 11	
PTR_BCS_UPCA_S	Not supported	
PTR_BCS_UPCE	1, 2, 3, 8, 9, 11	
PTR_BCS_UPCE_S	Not supported	

Table 139. POS printer BarCode symbologies supported (continued)

BarCode Symbology	JavaPOS and OPOS Gateway	OPOS
PTR_BCS_UPCD1	Not supported	
PTR_BCS_UPCD2	Not supported	
PTR_BCS_UPCD3	Not supported	
PTR_BCS_UPCD4	Not supported	
PTR_BCS_UPCD5	Not supported	
PTR_BCS_EAN8	Not supported	
PTR_BCS_JAN8	1, 2, 3, 8, 9, 11	
PTR_BCS_EAN8_S	Not supported	
PTR_BCS_EAN13	1, 2, 3, 8 ,9, 11	
PTR_BCS_JAN13	Not supported	
PTR_BCS_EAN13_S	Not supported	
PTR_BCS_EAN128	Not supported	
PTR_BCS_TF	Not supported	
PTR_BCS_ITF	1, 2, 3, 8 ,9, 11	
PTR_BCS_Codabar	1, 2, 3, 8 ,9, 11	
PTR_BCS_Code39	1, 2, 3, 8 ,9, 11	
PTR_BCS_Code93	1, 2, 3, 8 ,9, 11	
PTR_BCS_Code128	1, 2, 3, 8 ,9, 11	
PTR_BCS_OCRA	Not supported	
PTR_BCS_OCRB	Not supported	
PTR_BCS_Code128_Parsed	1, 2, 3	
PTR_BCS_RSS14	Not supported	
PTR_BCS_RSS_EXPANDED	Not supported	
PTR_BCS_PDF417	1, 2, 3 (Receipt only)	
PTR_BCS_MAXICODE	Not supported	
PTR_BCS_OTHER	Not supported	

In the following table, the specific models and stations to which escape sequences can be sent are listed in parentheses () after the type identifier. If none are specified, assume that all models and stations can accept the escape sequence. An application can also determine if an escape sequence can be sent to a particular printer station by calling the validateData method.

Table 140. POS printer escape sequences

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
Alternate color	ESC lrC	1, 2, 3 (TI3-5, Tx6-9) EC > 0x33 with Configured setting and correct paper (receipt)	1, 2, 3 (TI3-5, Tx6-9) EC > 0x33, 2CR, 2NR, 1NR with Configured setting and correct paper

Table 140. POS printer escape sequences (continued)

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
Alternate color (Custom)	ESC |#rC	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 with Configured setting and correct paper (receipt)	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33, 2CR, 2NR, 1NR with Configured setting and correct paper
Bold	ESC |(!)bC	Same as CapXxxBold (For Slp it must be used at the beginning of the data and applies to the entire printed line.)	Same as CapXxxBold
Center	ESC |cA	All	
Double high	ESC |3C	Same as CapXxxDhigh	
Double high & wide	ESC |4C	Same as CapXxxDwideDhigh	
Double wide	ESC |2C	Same as CapXxxDwide	
Feed and paper cut	ESC |#fP	All (receipt), except 13	All
Feed, paper cut, and stamp	ESC |#sP	• 1, 2, 3, 6, 11 (receipt) Prints downloaded receipt bitmap 1 • 7, 8, 9 (receipt) Prints the stamp stored with the DirectIO SET_STAMP_ID	1, 2, 3, 6, 7, 8
Feed lines	ESC |#lF	All	
Feed reverse	ESC |#rF	4, 5, 8, 11, 13 - Cannot mix with feed forward (receipt)	4, 5 - Cannot mix with feed forward
Feed units	ESC |#uF	All	
Fire stamp	ESC |sL	• 1, 2, 3, 11, 13 - Prints downloaded receipt bitmap 1 • 8, 9 - Prints the stamp stored with the DirectIO SET_STAMP_ID	1, 2, 3 - Prints downloaded receipt bitmap 1 6, 7, 8 - Supported Stamp, but not as a single command
Font typeface selection	ESC |#fT	Not supported	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33, 2CR, 2NR, 1NR with downloaded proportional font
Integrated escape	ESC |#E	Not supported	1, 2, 3
Italic	ESC |iC	Not supported	
Left Justify	ESC |1A	All	
Normal	ESC |N	All	
Paper cut	ESC |#P	All except 10, 12, 13	
Print bitmap	ESC |#B	• 1, 2, 3, 4, 5, 11, 13 • 7, 8, 9 (receipt)	1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13
Print bottom logo	ESC |bL	All	
Print in-line barcode	ESC |#R		Same as CapXxxBarcode
Print top logo	ESC |tL	All	

Table 140. POS printer escape sequences (continued)

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
Reverse video	ESC l(!)rvC	• 1, 2, 3 (receipt) • 7, 8, 9	1, 2, 3, 8, 11 (11 reverts to normal rotation in 180 mode)
RGB color	ESC l#fC	Not supported	
Right justify	ESC lrA	All	
Scale horizontally	ESC l#hC	1, 2, 3, 4, 5, 6, 7, 8, 9. Up to 2. Same as CapXxxDwide 1, 2, 3 (Tl3-5, Tx6-9) EC > 0*x*33 Up to 8 (receipt) 11: Up to 6 (receipt) 13: Up to 2	1, 2, 3, 4, 5, 6, 7, 8, 11. Up to 2. Same as CapXxxDwide 1, 2, 3 (Tl3-5, Tx6-9) EC > 0*x*33, 2CR, 2NR, 1NR, Up to 8
Scale vertically	ESC l#vC	1, 2, 3, 4, 5, 6. Up to 2 is for slip; up to 8 is for receipt. Same as CapXxxDhigh 1, 2, 3 (Tl3-5, Tx6-9) EC > 0*x*33 Up to 8 (receipt) 8, 9. Up to 2 11: Up to 6 (receipt) 13: Up to 2	1, 2, 3, 4, 5, 6, 7, 8, 11. Up to 2 is for slip; up to 8 is for receipt. Same as CapXxxDhigh 1, 2, 3 (Tl3-5, Tx6-9) EC > 0*x*33, 2CR, 2NR, 1NR Up to 8
Shading	ESC l#sC	7	7, 8
Single high and wide	ESC l1C	All	
Subscript	ESC ltbC	Not supported	
Superscript	ESC ltpC	Not supported	
Underline	ESC l(!)#uC	Same as CapXxxUnderline	

JavaPOS configuration

AddLineFeed

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.AddLineFeed" type="Boolean" value="True"/>
```

This property determines whether to add a line feed to the end of any text sent in a `printNormal`, `printImmediate`, or `printTwoNormal` command when it does not contain a complete line.

A complete line is defined as:

- A line finishing with the line feed character, for example:
  `"Complete Line\n"`
- A line finishing with an escape sequence that produces a line feed, for example:
  `"Complete Line ESC|10lF"`

- A line that completely fills the characters per line for the station in use, for example when RecLineChars=20 and the print text is:

  ```
  "Line of 20 chars7890"
  ```

Valid values are TRUE or FALSE. Default is TRUE.

Default8LPI

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.Default8LPI" type="String" value="False"/>
```

This property selects the font and the spacing needed to allows the user to print 8 lines per inch in the paper.

Valid values are TRUE and FALSE. Default is FALSE.

DefaultLargeFont

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.DefaultLargeFont" type="String" value="False"/>
```

This property selects the font with the largest size available on the printer, typically 12x24 dots on the receipt station and 7x9 dots on slip station.

Valid values are TRUE and FALSE. Default is FALSE.

DocAdvance

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.DocAdvance" type="String" value="50"/>
```

This property selects the number of motor steps to advance the paper on a Document Insert, from the top Document sensor to the first print position.

Valid values are 1 to 255. The default is 50.

Device support: 4610 TI-3/4/5/8/9.

DoubleHighAndDoubleWideIsQuad

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.DoubleHighAndDoubleWideIsQuad"
type="String" value="OFF"/>
```

This property enables the conversion of the double wide (1b|2C) and double high (1b|3C) escape sequences to work as double wide and high (1b|4C).

Only supported on 4610 Tx5/Tx7 using a DBCS character set.

Valid values are ON and OFF. Default is OFF.

PDF417AspectHeight

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.PDF417AspectHeight" type="String" value="1"/>
```

This property selects the value for the aspect ratio height of the PDF417 barcode.

Valid values are 1 to 9. Default is 1.

PDF417AspectWidth

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.PDF417AspectWidth" type="String" value="2"/>
```

This property selects the value for the aspect ratio width of the PDF417 barcode.

Valid values are 1 to 9. Default is 2.

PDF417ECCLevel

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.PDF417ECCLevel" type="String" value="15"/>
```

A security level is appended to the printed barcode. This property enables scanners to read the bar code even if it has been torn, written on, or otherwise damaged.

You can select a level from 0 to 8 to apply error correction. At level 0, a damaged PDF417 cannot be read, but the damage can be detected. At levels 1 through 8, a PDF417 symbol can still be read, even when damaged. As the error correction level increases, more damage can occur to the symbol and still be read. For values of 9 to 400, the ECC level is assume as a percentage of the code words in the barcode.

Valid values are 0 to 400. Default is 15.

PDF417Truncation

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.PDF417Truncation" type="String" value="OFF"/>
```

This property enables the truncation of the PDF417 barcode. The truncated PDF generates a single width bar symbol to replace the right row indicator and stop pattern. When truncation is enabled, decode performance is slightly degraded in order to allow more data to fit in the image width.

Valid values are ON or OFF. Default is OFF.

ProportionalFontFixedWidth

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.ProportionalFontFixedWidth"
type="String" value="20"/>
```

This property aligns proportional font characters on a fixed width. The value is expressed in printer dots. When this property is used, the proportional characters are treated as fixed characters.

Valid values are 8 to 32. Default is 20.

TranslateCharacter

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.TranslateCharacter.<AsciiHexCharNumber>"
type="String" value="<NewHexCharacter>"/>
```

This property replaces the value of the <*AsciiHexCharNumber*> parameter with the value of the <*NewHexCharacter* parameter>. In the text that prints, all the characters that match the AsciiHexCharNumber are replaced by the new character.

Values accepted:

- *<AsciiHexCharNumber>*: 0x00 to 0xFF in Hexadecimal representation. This value represents the character number in the ASCII table to be replaced.
- *<NewHexCharacter>*: The character to be used in place of the ASCII Character.

Examples:

- Replace the character 29 0x1D (also known as group separator) with the character G 0x47:

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.TranslateCharacter.0x1D" type="String" value="0x47"/>
```

- Replace the character K 0x4B with the character N 0x4E;

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.TranslateCharacter.0x4B" type="String" value="0x4E"/>
```

SureOneVersion

```
<prop name="SureOneVersion" type="String" value="Standard"/>
```

SureOneVersion defines the character set to be used by the service implementation of the SureOne printer thermal station. No default value is set, so you must indicate a selection.

To determine which character set is currently installed in the printer, perform an offline test as follows:

1. Turn off the system.
2. Press and hold the paper feed button.
3. Turn on the system and wait for a beep.
4. Release the paper feed button.

Possible values for this property are shown in the following table.

Table 141. SureOne Version property values

Possible Value	Character set
"Standard"	character set 998 (equal to character set 437)
"ChineseTrad"	character set 950
"ChineseSimp"	character set 1381
"Korean"	character set 1361
"Japanese"	character set 932

JavaPOS DirectIO calls

There are several DirectIO commands for downloading fonts to the PosPrinter. The general syntax for DirectIO commands is:

Syntax directIO (command: *int32*, **inout data:** *int32*, **inout obj:** *object* **): void { raises-exception }**

The device must be claimed and enabled before invoking the directIO() method.

To access POS printer constants, import: `com.ibm.jpos.services.DirectIO`.

Note: Note: The following directIO calls supported in IBM UnifiedPOS 1.7.5 might be deprecated in future releases:

- DirectIO.PTR_PD_LEFTTORIGHT
- DirectIO.PTR_PD_BOTTOMTOTO
- DirectIO.PTR_PD_RIGHTTOLEFT
- DirectIO.PTR_PD_TOPTOBOTTOM
- DirectIO.PTR_PM_PAGEMODE
- DirectIO.PTR_PM_NORMAL

FontInfo Object

The DirectIOFontInfo object is used as input or output to the DirectIO functions for downloading fonts to IBM 4610 and 4689 printers. The DirectIOFontInfo object has the structure shown in Table 142.

Table 142. DirectIOFontInfo object structure

Field Name	Type	Description
station	int	Print station indentifier: • POSPrinterConst.PTR_S_RECEIPT • POSPrinterConst.PTR_S_SLIP
codepage	int	Font number. This can have a value of 1 – 6.
filename	string	Font file name. This must be a fully qualified path.

Specific usage of the DirectIOFontInfo object will be documented for each DirectIO function

Get Font Info ID Command

Get Font information. Font information will be returned in the `com.ibm.jpos.services.sdiibmprinter.DirectIOFontInfo` object.

Table 143. DirectIO.GET_FONT_INFO_ID

Parameter	Type	Description
Command	Int32	com.ibm.jpos.services.DirectIO.GET_FONT_INFO_ID
Data[1]	Int32	com.ibm.jpos.services.DirectIO.GET_FONT_INFO
Obj	Object	com.ibm.jpos.services.sdiibmprinter.PrintGetDirectIOFontInfo

Remarks

None Defined

Errors None Defined

Download Prop Font ID Command

Download a proportional font to the IBM 4610 printer.

Table 144. DirectIO.DOWNLOAD_PROP_FONT_ID

Parameter	Type	Description
Command	Int32	com.ibm.jpos.services.DirectIO.DOWNLOAD_PROP_FONT_ID
Data	Int32	Not used
Obj	Object	com.ibm.jpos.services.sdiibmprinter. DirectIOFontInfo.

Remarks

Proportional fonts are supported only for the Receipt station. Valid values for the codepage field of DirectIOFontInfo are:

- 1: User-defined (UD) Code Page 1
- 3: UD Code Page 3

Notes:

1. Proportional font characters should have width and height between 8 and 32 inclusive.
2. When using proportional fonts, there is not a way to accurately wrap lines since each character may have a different width. You should manage the Line width using the RecLineChars property

After a successful font download, use the characterSet property to retrieve and set the desired code pages for font printing. Table 145 shows the usage of the user-defined character sets.

Table 145. User-defined (UD) character sets

Character set	Font attribute	CR station	DI station
101	Fixed	UD Code Page 1	UD CodePage 1
	Proportional	UD Code Page 1	N\A
102	Fixed	UD Code Page 2	N\A
	Proportional	N/A	N\A
103	Fixed	UD Code Page 3	UD CodePage 2
	Proportional	UD Code Page 3	N\A
104	Fixed	UD Code Page 4	N\A
	Proportional	N\A	N\A

Errors A UposException might be thrown when this method is invoked. Some possible values of the exception ErrorCode property are:

Table 146. UposException error codes

Value	Meaning
E_ILLEGAL	One of the following errors has occurred: • This printer does not support proportional fonts downloading. • This printer station does not allow proportional fonts. • Incorrect font number. • Incorrect font file type. • Character size not valid.
E_EXIST	Memory sector already has a font, erase the memory sector before downloading a font.
E_NOEXIST	Font file was not found.

Download Non-Proportional Font ID Command

Download non-proportional fonts to the IBM 4610 and 4689 printers.

Table 147. DirectIO.DOWNLOAD_NON_PROP_FONT_ID

Parameter	Type	Description
Command	Int32	com.ibm.jpos.services.DirectIO.DOWNLOAD_NON_PROP_FONT_ID
Data	Int32	Not used
Object	Object	com.ibm.jpos.services.sdiibmprinter. DirectIOFontInfo

Remarks for IBM 4689 POS Printer

- The *station* field of DirectIOFontInfo is ignored; the downloaded font applies to both the receipt and journal station.
- The *codepage* field of DirectIOFontInfo is ignored.
- For the IBM 4689 POSPrinter, non-proportional font characters for receipt/journal stations must be either width 8 and height 12 or width 12 and height 24.

Remarks for IBM 4610 POS Printer

- Valid values for the station field of DirectIOFontInfo are:
  - PTR_S_RECEIPT: Customer Receipt (CR) station
  - PTR_S_SLIP: slip (Document Insert or DI) station
- Valid values for the codepage field of DirectIOFontInfo are:
  - 1 – 4: one of the four codepages for the customer receipt (CR) station
  - 5 – 6: one of the two codepages for slip (DI) station
- Non-proportional font characters for the receipt station must have a width between 8 and 16 inclusive and a height between 16 and 32 inclusive; those for the slip station should have a width and height between 7 and 16 inclusive.

After a successful font download, use the characterSet property to retrieve and set the desired code pages for font printing. Table 145 on page 168 shows the usage of the different user-defined character sets.

Errors A UposException might be thrown when this method is invoked. Some possible values of the exception ErrorCode property are listed in Table 148:

Table 148. UposException error codes

Value	Meaning
E_ILLEGAL	One of the following errors has occurred: • This printer does not support **non**-proportional fonts downloading. • This printer station does not allow **non**-proportional fonts. • Incorrect font number. • Incorrect font file type. • Character size not valid.
E_EXIST	Memory sector already has a font, erase the memory sector before downloading a font.
E_NOEXIST	Font file was not found.

Download DBCS Font ID Command

Download double-byte character sets (DBCS) and fonts to a double-byte printer.

Table 149. DirectIO.DOWNLOAD_DBCS_FONT_ID

Parameter	Type	Description
Command	Int32	com.ibm.jpos.services.DirectIO.DOWNLOAD_DBCS_FONT_ID
Data	Int32	Not used
Object	Object	com.ibm.jpos.services.sdiibmprinter.DBCSFontInfo. Class that holds the filename of the DBCS font to download.

Remarks

None Defined

Errors A UposException might be thrown when this method is invoked. Some possible values of the exception ErrorCode property are listed in Table 150:

Table 150. UposException error codes

Value	Meaning
E_ILLEGAL	One of the following errors has occurred: • This printer does not support DBCS font downloads. • This printer station does not allow DBCS fonts. • Font file type is incorrect.
E_NOEXIST	Font file was not found.
E_FAILURE	Error reading from font file.

Additional JavaPOS information

The device service formats the entire print line internally. This includes left, center, and right alignments. It also inserts line breaks as necessary for long lines. This implementation approach does not allow a single print line to be built up from multiple printNormal() invocations.

The *LineHeight properties are not modifiable.

It is not possible to mix bold and normal printing on the Slip station of the 4610 printer. The line prints either all normal or all bold, depending on which escape sequence is used first.

Calibration for low paper sense

Low paper sensing on the 4610 printer model 2xR, 1NR based upon the paper thickness and core size. The user can modify the low paper and critically low sensing settings. These settings determine the amount of paper that remains on the roll and at which time the low-paper and critically low-paper status is sent. The default setting for low paper status is 5 meters and critically low status is 1 meter remaining on the roll.

The user must calibrate the printer to accurately recognize the thickness of the paper supply. The user only needs to calibrate one printer for a specific paper supply. They can then broadcast that calibration constant to all printers that will use the same paper. For details, refer to User's Guide for SureMark Printers Models 2xR/1NR.

1. Finding MCT values for low-paper sense by running offline test on the printer

- Add a new paper roll to be calibrated.
- Power off the printer.
- Hold down the CR button and turn on the printer.
- Release the CR button when CR light starts to flashing. A menu will be printed.
- Select option 5 – More selections
- Select option 1 – MCT listing
- Record the values for MCTs 0x26, 0x27, 0x28

2. Transfer the MCT values to aip46mct.cfg file located in the following directory:
   - Linux: /usr/share/pos/config/aipmct.cfg
   - Windows: c:\pos\config\aip46mct.cfg

   The mapping between MCT values and constants provided in 46mct.cfg file are as follows:
   - lowPaperSensing = MCT 0x26
   - criticallyLowPaperSensing = MCT 0x27
   - calibrationConstant = 0x28
3. To broadcast this value to other printers, transfer aip46mct.cfg file to clients to the location specified above and reboot the clients.

4610 printer firmware update

The IBM JavaPOS driver has the ability to automatically update the printer firmware if the printer is attached to the terminal when the terminal boots. It does not have this ability if the printer is attached to the terminal after the terminal has booted and the operating system has loaded.

In addition, printer firmware can be update manually at any time.

Updated printer firmware update files can be downloaded from the IBM website at: http://www.ibm.com/solutions/retail/store/support/

1. Go to: http://www.ibm.com/solutions/retail/store/support
   a. Under Peripherals, click **IBM SureMark Printer**.
   b. Under Downloads, select the appropriate link for your printer model.
   c. From the Search Results displayed, select **IBM 4610 SureMark Printer, Microcode Files for Firmware Update**.
   d. Download the zip file for `OPOS/JavaPOS`. This zip files contains multiple firmware files; choose the correct one for your printer type.
2. Place the firmware file for your printer type into the correct folder.

Table 151.

Operating system	Location
Windows	\POS\IBM\IBMJPOS\FLASH\RS485
Linux	/opt/ibm/javapos/flash/rs485

3. Make sure the printer is attached and restart.

Each model of 4610 printer has a different firmware file associated with it.

Table 152. IBM 4610 printer firmware update

Printer Model	Firmware update filename
TI1, TI2	aip46mc.hex
TI3, TI4, TG3, TG4, TF6, TM7	aip46mch.hex
TI5, TM7, TF7	aip46mcd.hex
TI8	aip46ti8.hex
TI9	aip46ti8.hex
2CR, 2NR	aip46v4.hex

If the firmware file you select does not match the type of printer you have attached, the program will not update the printer.

Automatic update

When a terminal starts, JavaPOS will check the firmware version, and update if necessary, all devices (including printers) that are defined in the JPOS.XML file.

On Windows, the firmware update process is controlled by a batch file, `\pos\ibmjpos\lib\flash.bat`. The batch file is executed automatically at startup by an entry in the registry, `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`.

Since the batch file controls the updating of firmware in all devices, not just printers, it is recommended that if you do not want the driver to automatically update the printer firmware then you should delete the firmware update files from the `\pos\ibmjpos\flash\rs485` folder.

On Linux, the firmware update process is controlled by a shell script, `/etc/init.d/flashdev`.

Since the batch controls the updating of firmware in all devices, not just printers, it is recommended that if you do not want the driver to automatically update the printer firmware then you should delete the firmware update files from the `/opt/ibm/javapos/flash/rs485` folder.

Manual update

Use this command line utility to manually update the firmware on an IBM 4610 printer. The syntax for the utility is:

```
java com.ibm.jpos.util.flash.AipFlash46s [param] [RS485_flash_filename]
```

where:

param:

```
-e: Update the printer microcode
```

RS485_flash_filename: The fully qualified path for the firmware file.

For example:

```
java com.ibm.jpos.util.flash.AipFlash46s –e /opt/ibm/javapos/flash/aipmch46.hex
```

4610 printer font download

A command utility is also provided to download fonts to the printer. The syntax for the utility is:

```
java com.ibm.jpos.util.font.AipFnt46s [param] [filename] [codepage]
```

where:

param:

- -f: fixed font file
- -p: proportional font file
- -d: dbcs font file (valid only for double-byte printers)

filename: The fully qualified pathname of the font file.

codepage:

- 1: CR (receipt) station Code Page 1
- 2: CR (receipt) station Code Page 2
- 3: CR (receipt) station Code Page 3
- 4: CR (receipt) station Code Page 4
- 5: DI (slip) station Code Page 1
- 6: DI (slip) station Code Page 2
- 0: DBCS font download

For example:

```
java com.ibm.jpos.util.font.AipFnt46s -p /opt/ibm/javapos/fonts/FontFile.fTH 3
```

Using POSPrinter's MICR device

When using a printer with an integrated MICR, it is possible for a multi-threaded application to attempt to talk to both devices at the same time. Unpredictable results can occur. Such applications should provide a locking mechanism to ensure that only one thread at a time talks to these devices.

Handling invalid characters

The first 31 ASCII characters are non-printable. The POSPrinter handles these as strings and prints the hexadecimal value of the character in the string. Table 153 shows the values that the POSPrinter prints for these values.

Table 153. Handling invalid characters

Hex Value	Character	Also known as	JavaPOS print
0x00	NUL	Null	00
0x01	SOH	Start of Heading	01
0x02	STX	Start of Text	02
0x03	ETX	End of Text	03
0x04	EOT	End of Transmission	04
0x05	ENQ	Enquiry	05
0x06	ACK	Acknowledge	06
0x07	BEL	Bell	07
0x08	BS	Backspace	08

Table 153. Handling invalid characters (continued)

Hex Value	Character	Also known as	JavaPOS print
0x09	Tab	Horizontal Tab	*
0x0A	LF	NL line feed, New Line	*
0x0B	VT	Vertical Tab	0B
0x0C	FF	NP form feed, New Page	0C
0x0D	CR	Carriage Return	*
0x0E	SO	Shift Out	0E
0x0F	SI	Shift In	0F
0x10	DLE	Data Link Escape	10
0x11	DC1	Device Control 1	11
0x12	DC2	Device Control 2	12
0x13	DC3	Device Control 3	13
0x14	DC4	Device Control 4	14
0x15	NAK	Negative Acknowledge	15
0x16	SYN	Synchronous Idle	16
0x17	ETB	End of Transmission Block	17
0x18	CAN	Cancel	18
0x19	EM	End of Medium	19
0x1A	SUB	Substitute	1A
0x1B	ESC	Escape	*
0x1C	FS	File Separator	1C
0x1D	GS	Group Separator	1D
0x1E	RS	Record Separator	1E
0x1F	US	Unit Separator	1F

* These values are processed by the POSPrinter.

Escape sequence handling

- The Fire Stamp escape sequence is supported on the 4610 printers. JavaPOS uses the bitmap downloaded as bitmap #1 for the stamp.
- Handling of invalid, unrecognized, or illegal value escape sequences is performed as follows:
  - Sequences that do not start with `esc|` are passed to the printer unmodified. In this case, validateData() returns JPOS_E_FAILURE.
  - Sequences that start with `esc|` but are not valid JavaPOS sequences are passed through unmodified. In this case validateData() returns JPOS_E_FAILURE.
  - Sequences that specify a function not supported by the printer are ignored. Note that they are *not* passed to the printer. The validateData() would return JPOS_E_FAILURE.
  - Sequences that include a number that is considered invalid (such as less than 0, or in the case where the number represents a percentage, greater than 100), the device service chooses a number to use. This is an interpretation of the validateData method in the case when JPOS_E_ILLEGAL is returned. The

specification states that the service can select valid alternatives. In this case validateData() would return JPOS_E_ILLEGAL.

Line wrapping on the slip station using rotatePrint method(PTR_RP_LEFT90)

The hardware cannot determine the width of the slip station paper when print is rotated 90 degrees left; it is not possible to perform automatic wrapping in this case. Applications should not assume the printer service provides automatic wrapping before reaching the edge of the paper while in this mode. Set the SlpLineChars value property to a value that causes the print lines to wrap correctly for your document length.

Where printing is rotated 90 degrees on the slip station, only the following escape sequences are supported:

- Top and bottom log
- Double wide
- Left, center, and right alignment

Note: The 4610 hardware is not able to print characters from a font with a size of 16 x 16 dots on the slip station (document station) while in rotate mode.

Bitmap printing

To use setBitmap and printBitmap in a non-graphics environment on Linux, the following parameter must be passed to java:

```
java -Djava.awt.headless=true
```

- The methods printBitmap and setBitmap support JPEG and GIF file formats. Uncompressed Windows bitmaps (.BMP) up to 8 bits per pixel are also supported.
- If the bitmap width exceeds the RecLineWidth or SlpLineWidth, an exception is thrown with ErrorCodeExtended = EPTR_TOOBIG.

Table 154. Bitmap printing

Type of printer	Station	Method	Property	Max height	Size
4610 TI 3/4/5	receipt	printBitmap()	RecLineWidth	None	None
4610 TI 3/4/5	receipt	setBitmap()	RecLineWidth	2040	w*h/8<8K
4610 TI 3/4/5	slip	printBitmap()	SlpLineWidth	None	None
4610 TI 3/4/5	slip	setBitmap()	SlpLineWidth	40	w*h/8<8K

Note: printBitmap() has no height maximum because the bitmap is divided into horizontal slices. The height is physically limited on the slip station to the length of the paper.

Bitmap quality

UPOS standard has defined the RecLetterQuality and SlpLetterQuality properties to control the bitmap and text quality. Prior to IBM UnifiedPOS release 1.9.2, IBM drivers supported High Quality bitmap printing only.

For IBM UnifiedPOS 1.9.2 and later the driver follows the UPOS specification. When the LetterQuality property is false the bitmap will be printed in low quality/high speed mode, and when it is true, the bitmap will be printed in high quality mode.

If you want to force bitmaps to always print in high quality mode, you must add the following property in the POSPrinter's JposEntry:

```
<prop name="com.ibm.jpos.sdi.config.POSPrinter.BitmapLetterQuality" type="Boolean" value="true"/>
```

Values accepted:

True Use high quality mode always.

False (default)
Use UPOS RecLetterQuality/SlpLetterQuality setting to control the bitmap quality.

Device support:

4610 TI-3/4/5/8/9 TM/F 6/7

Supported since:

UPOS 1.9.5

Color support

Two-color printing is supported on the 4610 Models TI3, TI4, TI5, TI6, TI8, and TI9 printers that have a microcode EC level of 33 or greater. To enable color printing the JavaPOS configuration entry for the printer must be modified to have the following property:

```
<prop name="colorMode" type="Integer" value="2"/>
```

Any value other then 2 defaults to single color. Color mode should be configured only if color printing is used, because enabling the feature slows down the printer, whether or not color printing is performed.

Check flipping support

While the drivers still support the DirectIO.FLIP_CHECK_ID, the preferred method to flip the check is to use the POSPrinter method:

```
changePrintSide(  )
```

PageMode support

JavaPOS supports IBM PageMode on Receipt Station only. When in PageMode, use of transactionPrint and rotatePrint are prohibited and will cause an exception to be thrown. When horizontalPosition or verticalPosition is set in PageMode, if the value set is not supported by the printer, the UPOS driver adjusts the value to the closest supported value.

PageMode properties (default)

- AfterOpen (no station):
  - PageModeArea = ″″
  - PageModePrintArea = ″″
  - PageModeHorizontalPosition = 0
  - PageModeVeritcal Position = 0
  - PageModePrintDirection = 0
  - PageModeStation = 0
- After setting a valid station:
  - PageModeArea = ″576,800″ (″576,1250″ for T*x*8/T*x*9)
  - PageModePrintArea = ″0,0,0,0″
  - PageModePrintDirection = PTR_PD_LEFT_TO_RIGHT
- After entering PageMode:
  - PageModePrintArea = ″0,0,576,800″ (″0,0,576,1250" for T*x*8/T*x*9).
- Setting the PrintDirection: Property may change PageModeHorizontalPosition and PageModeVerticalPostion.
- If the Vertical or Horizontal Position property values are not supported, then they will be set to the closest supported value. If the values are outside of the area described by PageModePrintArea, the driver sets these values to the size of the area described by PageModePrintArea.

Note: The vertical and horizontal position properties must be set before sending any print command. If you set the vertical position near the top edge, then text, bitmaps, and barcodes do not display because they are beyond the PageMode area. Set the vertical position to be greater than the vertical dimension of the image to print.

Several examples below illustrate the use of PageMode printing.

Example: Not specifying the PageMode Vertical Position/PageMode Horizontal Position results in partial image printing.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModeArea("0,0,576,1250");
printer.pageModePrint(jpos.POSPrinterConst.PTR_PM_PAGE_MODE);
printer.printBitmap(POSPrinterConst.PTR_S_RECEIPT,"logo.bmp";
    POSPrinterConst.PTR_BM_ASIS;
    POSPrinterConst.PTR_PM_CENTER);
printer.pageModePrint(jpos.POSPrinterConst.PTR_PM_NORMAL);
```

Figure 26. Partial image, no position setting

In the next example the printing is correct, with position properties set.

Example:

This example illustrates how to print a bitmap and a barcode correctly.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModePrintArea("0,0,576,400");
printer.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);
printer.setPageModeHorizontalPosition(0);
printer.setPageModeVerticalPosition(150);
printer.printBitmap( POSPrinterConst.PTR_S_RECEIPT,"logo.bmp",
    POSPrinterConst.PTR_BM_CENTER;
    POSPrinterConst.PTR_BM_ASIS);
printer.setPageModeHorizontalPosition(0);
printer.setPageModeVerticalPosition(300);
printer.printBarCode( POSPrinterConst.PTR_S_RECEIPT,"01234567",
    108,
    POSPrinterConst.PTR_BCS_CODE39,
        100, 200,
    POSPrinterConst.PTR_BC_CENTER;
    POSPrinterConst.PTR_BM_CENTER);
printer.pageModePrint(jpos.POSPrinterConst.PTR_BC_TEXT_BELOW;
```

Figure 27. Setting position properties for printing

Example: This is another example to illustrate how to print a bitmap.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModePrintArea("0,0,576,400");
printer.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);
printer.setPageModeHorizontalPosition(0);
printer.setPageModeVerticalPosition(150);
printer.printBitmap( POSPrinterConst.PTR_S_RECEIPT,"logo.bmp",
     POSPrinterConst.PTR_BM_CENTER;
     POSPrinterConst.PTR_BM_ASIS);
printer.pageModePrint(jposPOSPrinterConst.PTR_PM_NORMAL);
```

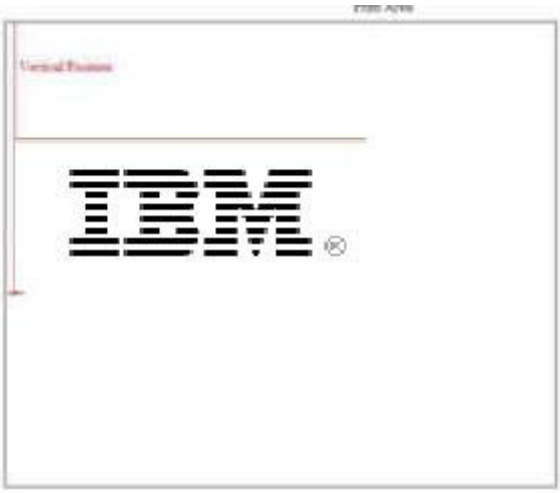


Figure 28. Setting image position for printing

Example: This example illustrates printing two lines of code in PageMode.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModePrintArea("0,0,576,400");
printer.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);
printer.printNormalText(POSPrinterConst.PTR_S_RECEIPT,"Text");
printer.printNormalText(POSPrinterConst.PTR_S_RECEIPT,"TextSecondLine");
printer.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

Printing text: Begin in the next line.

Figure 29. Printing text

When an image follows printed text, the image appears larger than when the image is printed alone, because the image begins in the next line of the text.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModePrintArea("0,0,576,400");
printer.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);
printer.printNormal( POSPrinterConst.PTR_S_RECEIPT,"TestingPrintText\n");
printer.printBitmap( POSPrinterConst.PTR_S_RECEIPT,"logo.bmp";
     POSPrinterConst.PTR_BM_CENTER;
     POSPrinterConst.PTR_BM_ASIS)
printer.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

Figure 30. Image follows text

Example:

When you print an image or barcode, the following text, image, or barcode prints next to the first.

Code:

```
printer.setPageModeStation(POSPrinterConst.PTR_S_RECEIPT);
printer.setPageModePrintArea("0,0,576,1250");
printer.pageModePrint(jpos.POSPrinterConst.PTR_PM_PAGE_MODE);
printer.printBarcode( POSPrinterConst.PRT_S_RECEIPT,"01234567", 108, 100,
        200, -1, -13);
printer.printBitmap( POSPrinterConst.PTR_S_RECEIPT,"logo.bmp",
    POSPrinterConst.PTR_BM_CENTER;
    POSPrinterConst.PTR_BM_ASIS);
printer.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

Figure 31. Second image beside first image

SetPrintArea: If an empty string buffer is sent as a parameter to PageModePrintArea, and a get is performed, the current PrintArea is returned. If non-numeric values are used, or the wrong number of values are entered, an exception is thrown. The driver adjusts values to the printer supported values as shown below:

```
if x < 0 then x = 0
if y < 0 then y = 0
if x > MaxWidth - MinWidth then
    x = MaxWidth - MinWidth
if y > MaxHeight - MinHeight
    then y = MaxHeight - MinHeight
if width < MinWidth then width = MinWidth
if height < MinHeight then height = MinHeight
if (width + x) > MaxWidth then width = MaxWidth - x
if (height + y) > MaxHeight then height = MaxHeight - y
```

The values for MinHeight, MaxHeight, MinWidth, and MaxWidth are as follows:

- MinHeight: 6
- MaxHeight: 800 (1250 for TI8/9)
- MinWidthPosition: 15
- MaxWidthPosition: 576

Examples:

- ″0,-1,570,600″ is adjusted to: ″0,0, 570,600″
- ″-1,0,800,5000″ is adjusted to: ″0,0,576,800″ for TI8/9, 0,0,576,1250
- ″-1,6,200,8000″ is adjusted to: ″0,6,200,794″ for TI8/9, 0,6,200,1244
- ″″ gets the current value of PageModePrintArea

Function of the ErrorEvent with response of ER_CLEAR

When the application responds to an ErrorEvent with ER_CLEAR, it clears all outputs that have not been completed. For example:

1. open/ claim/ enable/ async=true
2. printNormal1
3. ErrorEvent on printNormal2
4. printNormal3
5. printNormal4
6. Respond ErrorEvent with ER_CLEAR
7. printNormal5

All the printNormal events up to, and including printNormal4, are cleared in step 6.

Concurrent MICR/full-image scan

The 4610 Model TI9 performs a concurrent MICR/full-image scan; however, when using a Model TI8 you must do separate passes.

Rec/SlpLineChars behavior

If the parameter for setRec/SlpLineChars method is negative, the method will throw a JPOS_E_ILLEGAL exception, if value is 0<=x<=10 the Rec/SlplineChars property will be set to 10.

OPOS configuration

Unsupported settings

The following settings are *unsupported and untested* options used to modify Service Object behavior only in very limited cases. Use them only when directed to do so by IBM Support.

Note: These settings cannot be modified with the Configuration tool.

Table 155. Service Object settings for RS-485/USB devices (POS printer), unsupported

Keyword	Type	Description	OPOS Gateway Support
AsyncBufferDelay	String	Number of milliseconds Async Thread will wait for Asynchronous request to be added to the queue before processing begins. The default value is 100.	No
AsyncBlockChase	String	Whether the printer driver waits for confirmation that the Asynchronous Block of data has been printed before continuing with print commands. Valid values are: **ON** Wait (default) **OFF** Do not wait	No
AsyncCarriageReturn	String	Whether a carriage return character is added after each print command in Async Mode. Valid values are: **ON** Add carriage return **OFF** Do not add carriage return (default)	No

Table 155. Service Object settings for RS-485/USB devices (POS printer), unsupported (continued)

Keyword	Type	Description	OPOS Gateway Support
SyncPrintWithoutWait	String	Whether the printer driver waits for confirmation that the synchronous data has been printed before continuing with print commands. Valid values are: **False** Wait (default) **True** Do not wait	No
ChineseDBCS	String	4610 only. Enables the use of bitmap printing to support DBCS characters. The character map is downloaded to the subsystem and converted to bitmap at print time. (No longer supported due to the availability of models TI5 and Tx7.) Valid values are: **ON** Enabled **OFF** Disabled (default)	No
StripDBLineFeeds	String	Whether carriage-return and line-feed characters are stripped out when in Chinese Double-Byte mode. Valid only when the ChineseDBCS setting is ON. Valid values are: **ON** Strip CR and LF **OFF** Do not strip (default)	No
OnlineTimeout	String	Time in milliseconds to wait for device to come online. Only applies to RS485/USB printers.	No
BarcodeScaleHeightCodabar	String	Value to scale height of Codabar barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightCode128	String	Value to scale height of Code128 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightITF	String	Value to scale height of ITF barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightJAN13	String	Value to scale height of JAN-13 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightJAN8	String	Value to scale height of JAN-8 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightUPCA	String	Value to scale height of UPC-A barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleHeightUPCE	String	Value to scale height of UPC-E barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthCodabar	String	Value to scale width of Codabar barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthCode128	String	Value to scale width of Code128 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthCode39	String	Value to scale width of Code39 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthITF	String	Value to scale width of ITF barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthJAN13	String	Value to scale width of JAN-13 barcodes. Only applies to RS485/USB printers.	No

Table 155. Service Object settings for RS-485/USB devices (POS printer), unsupported (continued)

Keyword	Type	Description	OPOS Gateway Support
BarcodeScaleWidthJAN8	String	Value to scale width of JAN-8 barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthUPCA	String	Value to scale width of UPC-A barcodes. Only applies to RS485/USB printers.	No
BarcodeScaleWidthUPCE	String	Value to scale width of UPC-E barcodes. Only applies to RS485/USB printers.	No
Default40CPL	String	Set the receipt to default to 40 characters per line. Does so by setting **InterCharSpacingReceipt12Cpi** to 2 dots. Valid values are TRUE and FALSE. Default is FALSE. Only applies to RS485/USB printers.	No
DocAdvance	String	Sets the number of motor steps from the top Document sensor to the first print position. Valid values are 1 to 256. Default is 50. Only applies to RS485/USB printers.	Yes
InterCharSpacingDocument12Cpi	String	Sets the number of printer dots between characters when the Slip station is set to 12 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingDocument15Cpi	String	Sets the number of printer dots between characters when the Slip station is set to 15 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingDocument17Cpi	String	Sets the number of printer dots between characters when the Slip station is set to 17 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingDocument20Cpi	String	Sets the number of printer dots between characters when the Slip station is set to 20 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingReceipt12Cpi	String	Sets the number of printer dots between characters when the Receipt station is set to 12 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingReceipt15Cpi	String	Sets the number of printer dots between characters when the Receipt station is set to 15 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingReceipt17Cpi	String	Sets the number of printer dots between characters when the Receipt station is set to 17 characters per inch. Only applies to RS485/USB printers.	No
InterCharSpacingReceipt20Cpi	String	Sets the number of printer dots between characters when the Receipt station is set to 20 characters per inch. Only applies to RS485/USB printers.	No
LegacyDIMode	String	In prior releases, the OPOS_SUE_PTR_PAPEROK was fired once the document was registered, not merely detected by the front sensor. This setting allows customers to maintain this behavior. Valid values are TRUE and FALSE. Default is FALSE. TRUE means the OPOS_SUE_PTR_PAPEROK event is fired when the paper is detected at the front sensor. FALSE means the OPOS_SUE_PTR_PAPEROK event is fired when the paper is registered. Only applies to RS485/USB printers.	No

Table 155. Service Object settings for RS-485/USB devices (POS printer), unsupported (continued)

Keyword	Type	Description	OPOS Gateway Support
LegacyRS232Mode	String	In prior releases, the OPOS_SUE_PTR_NEAREND was fired when the document was detected only by the Ready and Front Sensor. This behavior was specific to the RS232 printer. Valid values are TRUE and FALSE. Default is FALSE. TRUE means the OPOS_SUE_PTR_NEAREND event is fired when the document is positioned in the front. FALSE means that OPOS_SUE_PTR_NEAREND event is not fired when the document is positioned in the front. Only applies to RS485/USB printers.	No
RecLineSpacing	String	Sets the default RecLineSpacing. Default is 34 (dots). Only applies to RS485/USB printers.	No
TransactionHold	String	Determines whether to hold/release the buffer internal to the printer during a TransactionPrint to the receipt. Valid values are TRUE and FALSE. Default is FALSE. Only applies to RS485/USB printers.	No
BitmapResolution	String	Indicates bitmap resolution on SurePOS 100/SureOne thermal printers. Valid values are 0 (low) and 9 (high) only. Default is 9.	No
SetCompatibilityMode	String	This property allows the 2xR, 1NR printer to remain in compatibility mode even when driver is updated. Valid values are: True False (default) **Notes:** 1. If no property is found, the default value is used 2. If the property is present but it has an invalid value, then the default value is used	No

Supported settings

Note: Not all of these settings can be modified with the Configuration tool.

Table 156. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported

Keyword	Type	Description	OPOS Gateway Support
BitmapSpeedQuality	String	Speeds up bitmap downloading. Valid values are: **False** (default) **True**	No

Table 156. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported (continued)

Keyword	Type	Description	OPOS Gateway Support
PDF417ECCLevel	String	This property adds on the barcode a printed security level to enable scanners to read the bar code even if it has been torn, written on, or damaged in other ways. Error correction is specified by selecting a level from 0 to 8. At level 0, a damaged PDF417 cannot be read, but the damage can be detected. At levels 1 through 8, a PDF417 symbol can still be read, even when damaged. As the error correction level increases, more damage can occur to the symbol and still be read. For values of 9 to 400, the ECC level is assumed as a percentage of the code words in the barcode. **Valid values:** 0 to 400 in decimal representation. **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used: 15. 3. If the property is present but the value is not a number, the default value is used. 4. If the property is present but it has a value not in the range of values, it is matched to the closest parameter.	Yes
PDF417AspectHeight	String	This property selects the value for the aspect ratio height of the PDF417 barcode. **Valid values:** 1 to 9 in decimal representation. **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used: 1. 3. If the property is present but the value is not a number, the default value is used. 4. If the property is present but it has a value not in the range of values, it is matched to the closest parameter.	Yes
PDF417AspectWidth	String	This property selects the value for the aspect ratio width of the PDF417 barcode. **Valid values:** 1 to 9 in decimal representation. **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used: 2. 3. If the property is present but the value is not a number, the default value is used. 4. If the property is present but it has a value not in the range of values, it is matched to the closest parameter.	Yes

Table 156. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported (continued)

Keyword	Type	Description	OPOS Gateway Support
PDF417Truncation	String	This property enables the truncation of the PDF417 barcode. Truncated PDF generates a symbol with the right row indicator and stop pattern replaced by a single width bar. There is a slight degradation in decode performance if truncation is enabled, which allows more data to fit in the image width. **Valid values:** **ON** (Enables truncation) **Other value** (Disables truncation, default) **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used.	Yes
TranslateCharacter	Key	Key values for translating one character to another. This is a key folder in the Registry in which entries that can replace the characters are stored. See TranslateCharacter\<X> for details about the entries that can be added inside this key. **Note:** This setting cannot be modified with the configuration tool.	Yes
TranslateCharacter\<X>	String	The one-byte hexadecimal value of the character to translate. For example, when you translate char "E" (0x45) to char "e" (0x65), the entry name is 45 and the entry value is 65. Write *only* the number in the entry. **Note:** This setting cannot be modified with the Configuration tool.	Yes
CapRec2Color	String	When color thermal paper is loaded in the SureMark printer, the valid values are: **N** Color paper not loaded (default) **Y** Color paper loaded	Yes
ProportionalFontFixedWidth	String	This property aligns proportional font characters on a fixed width. The value is expressed in printer dots. If this property is used the proportional characters are treated as fixed characters. **Valid values:** 8 to 32 in decimal representation. **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used: 20. 3. If the property is present, but the value is not a number, the default value is used. 4. If the property is present but it has a value not in the range of values, it is matched to the closest parameter.	Yes

Table 156. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported (continued)

Keyword	Type	Description	OPOS Gateway Support
PrinterModel	String	4689 only. Specifies the model of 4689 printer in use. Valid values are: • 4689-TD5 • 4689-3M1 • 4689-3G1 • 4689-TD5(integrated into 4674) • 4689-TG1(integrated into 4674)	No
PersistantBitmaps	String	Enables storage of downloaded bitmap in the registry. Valid values are: **False** Disabled (default). Service Object must download the bitmap each time the driver is opened. **True** Enabled. Downloaded bitmap is stored in the registry. **Note:** This setting cannot be modified with the Configuration tool.	No
ErrorOnCoverOpen	String	Fired Error Event when cover is opened. Valid values are: **False** Do not fire event (default) **True** Fire event **Note:** This setting cannot be modified with the Configuration tool.	No
LegacyRS-232Mode	String	Processes same as prior to release 1.7.1. Valid values are: **OFF** Default **ON** **Note:** This setting cannot be modified with the Configuration tool.	No
LegacyDIMode	String	Processes DI same as prior to release 1.7.1. Valid values are: **OFF** Default **ON** **Note:** This setting cannot be modified with the Configuration tool.	No
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.	No
OEMEmulation	String	Emulate OEM printer. Valid values are: **False** (default) **True**	No

Table 156. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported (continued)

Keyword	Type	Description	OPOS Gateway Support
DefaultLargeFont	String	This property selects the font with the largest size available on the printer, typically: 12 dots x 24 dots on Receipt and 7 half dots x 9 dots on SLIP. The selection is made when the device is OPENed. Valid values are: **True** **False** (default) **Notes:** 1. If no property is found, the default value is used: False. 2. If the property is present but it has an invalid value, then the default value is used.	Yes
Default8LPI	String	This property selects the spacing needed to allow the user to print 8 lines per inch on the paper. Valid values are: **True** **False** (default) **Notes:** 1. If no property is found, the default value is used. 2. If the property is present but it has an invalid value, then the default value is used.	Yes
DocAdvance	String	This property selects the number of motor steps to advance the paper on a Document Insert, from the top Document sensor to the first print position.**Valid values:** 1 to 255 in decimal representation. **Notes:** 1. This setting cannot be modified with the Configuration tool. 2. If no property is found, the default value is used: 50 (0x32). 3. If the property is present but the value is not a number, the default value is used. 4. If the property is present but it has a value not in the range of values, it is matched to the closest parameter. 5. Applies only to EIA-232 4610 printer.	Yes
SetCompatibilityMode	String	This property allows the 2xR, 1NR printer to remain in compatibility mode even when driver is updated. Valid values are True, False (default) **Notes:** 1. If no property is found, the default value is used 2. If the property is present but it has an invalid value, then the default value is used.	No

Table 157. Service object settings for SureOne devices (POS printer)

Keyword	Type	Description	OPOS Gateway Support
ThermalPrinter	String	Whether printer is impact or thermal. Valid values are: **0** Impact (default) **1** Thermal **Note:** Applies only to SureOne/SurePOS 100.	No
NarrowPaper	String	Whether paper is normal or narrow width. Valid values are: **0** Normal (default) **1** Narrow **Note:** Applies only to SureOne/SurePOS 100.	No
BitmapResolution	String	Bitmap resolution (for thermal printers only). Valid values are: **0** Low resolution (default) **9** High resolution **Note:** Applies only to SureOne/SurePOS 100.	No
PrinterModel	String	Specifies the model of printer in use. Valid values are: **0** Single-head impact (default) **1** Double-head impact **2** Thermal **3** A04/A05 impact **Note:** Applies only to SureOne/SurePOS 100.	No

OPOS DirectIO calls

Miscellaneous

Flip Check

This flips the document currently in the DI station. It is supported on all printers that contain a check flipper.

Table 158. Flip check

Parameter	Type	Description
Command	Int32	0
Data	Long*	Ignored
Obj	BSTR*	Ignored

Attention: This OPOS DirectIO call should not be used. Applications should use the changePrintSide method of the printer object instead.

Print Downloaded Bitmap

This prints a bitmap previously stored in the printer. Not supported on EIA-232 printers.

Table 159. Print downloaded bitmap

Parameter	Type	Description
Command	Int32	0

Table 159. Print downloaded bitmap (continued)

Parameter	Type	Description
Data	Long*	Pointer to a byte containing the bitmap number
Obj	BSTR*	Ignored

Attention: This OPOS DirectIO call should not be used. Applications should use the SetBitmap method and the ESC|#B escape sequence of the printer object instead.

Print Downloaded Message

This prints a message previously stored in the printer. It is not supported on EIA-232 printers.

Table 160. Print downloaded message

Parameter	Type	Description
Command	Int32	0
Data	Long*	Pointer to a byte containing the message number
Obj	BSTR*	Ignored

Read MICR Data

This reads the MICR data from the document currently in the DI station. It is only supported on printers connected via USB and RS-485.

Table 161. Read MICR data

Parameter	Type	Description
Command	Int32	4
Data	Long*	Ignored
Obj	BSTR*	Ignored

Attention: This OPOS DirectIO call should not be used. Applications should use the MICR service object instead.

Reset Printer Object

This forces the service object to reset itself to a known state. It is only supported on printers connected via USB and RS-485.

Table 162. Reset printer object

Parameter	Type	Description
Command	Int32	501
Data	Long*	Ignored
Obj	BSTR*	Ignored

Check for Errors

This forces the service object to determine if the printer is in an error condition. It will cause the ResultCode and ResultCodeExtended properties to be reset and is only supported on printers connected via USB and RS-485.

Table 163. Check for errors

Parameter	Type	Description
Command	Int32	502
Data	Long*	Ignored
Obj	BSTR*	Ignored

Set Double Strike Mode

This sets double-strike mode. It is only supported on Model 4 printers.

Table 164. Set double strike mode

Parameter	Type	Description
Command	Int32	20
Data	Long*	Pointer to a byte which determines the double-strike status Valid values are: 0 (disable) 1 (enable)
Obj	BSTR*	Ignored

Flash memory

All OPOS access to the Flash memory in the SureMark printer is through DirectIO() methods. The following DirectIO commands are added to the SureMark OPOS Service Object for the RS-485, USB and EIA-232 interfaces.

Table 165. Flash memory

Parameter	Type	Description
Command	Int32	FLASH_MEMORY
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

Write a record to flash memory. The format of the data to be written is r1.r2.r3.r4.n1.n2.data, where r1,r2, r3 and r4 are 32-bit numbers, in little-endian format, indicating the record number and n1 and n2 are 16-bit numbers, in little-endian format, indicating the number of data bytes to be written.

When AsyncMode is set to true, the data is queued to the printer. Any error associated with this write causes a DirectIoEvent. If the write is successful, an OutputCompleteEvent is fired.

If the record number is beyond the maximum records supported, pData is set to DIRECTIO_FLASH_ERROR_REASON_OUT_OF_RANGE (0x2F).

If the record number is beyond the record length or is longer than the set record length or the maximum supported, pData is set to DIRECTIO_FLASH_ERROR_REASON_TOO_LONG (0x2D).

When AsyncMode is false, this command waits until the memory is written to the printer (or an error condition occurs) before returning control to the application. If there is an error, ResultCodeExtended is set with either of the two values, or with the POSSWIN internal error.

Read flash memory

Table 166. Read flash memory

Parameter	Type	Description
Command	Long	FLASH_MEMORY (0x11)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

Read the flash memory record number. The format of the data to be written is r1.r2.r3.r4.n1.n2.data, where r1, r2, r3 and r4 are 32-bit numbers, in little-endian format, indicating the record number.

When AsyncMode is set to true, the data is returned to the application by a DirectIoEvent with the EventNumber set to DIRECTIO_FLASH_DATA (0xFD). Any error associated with this read causes a DirectIoEvent with the EventNumber set to DIRECTIO_FLASH_ERROR (0xFE).

If the record number is beyond the maximum records supporte, pData is set to DIRECTIO_FLASH_ERROR_REASON_OUT_OF_RANGE (0x2F).

If there is not response to the request within 5 seconds, pData is set to DIRECTIO_FLASH_ERROR_REASON_CMD_TIMEOUT (0x2E).

If the record number is beyond the record length or is longer than the set record length or the maximum supported, pData is set to DIRECTIO_FLASH_ERROR_REASON_TOO_LONG (0x2D).

When AsyncMode is false, this command waits until the memory is read from the printer (or an error condition) before returning the data to the application. The record is returned in the pString field of the DirectIO call. If there is an error, ResultCodeExtended is set with any one of these three values, or with the POSSWIN internal error.

Query flash size

Table 167. Query flash size

Parameter	Type	Description
Command	Long	QUERY_FLASH_SIZE (0x12)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Returns the size of memory.

The value is returned in the pData field of the DirectIO call.

Query max records

Table 168. Query max records

Parameter	Type	Description
Command	Long	QUERY_MAXIMUM_RECORDS (0x13)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Returns the maximum number of records. This number is calculated by dividing the maximum printer memory by the application requested memory size.

The value is returned in the pData field of the DirectIO call.

Set record length

Table 169. Set record length

Parameter	Type	Description
Command	Long	SET_RECORD_LENGTH (0x14)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Specifies the number of bytes for each record written to the flash memory. The flash memory should be erased after changing the record size. The OPOS drivers do not automatically erase the memory.

Erase flash memory

Table 170. Erase flash memory

Parameter	Type	Description
Command	Long	ERASE_FLASH_MEMORY (0x15)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Erases all data stored in the flash memory on the 4610 printer.

Get record length

Table 171. Get record length

Parameter	Type	Description
Command	Long	GET_RECORD_LENGTH (0x16)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Retrieves the number of bytes for each record written to the flash memory.

The value is zero if it has not yet been set after an erase.

Check scanning support

DirectIO commands are available to support check scanning on the SureMark™ TI8/9 POS Printers. The supported commands for DirectIO check scanning are in the sections below.

Micro read with scan

Table 172. Micro read with scan

Parameter	Type	Description
Command	Long	CS_MICR_READ_WITH_SCAN (201)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

This command is used to configure the printer to perform a MICR read in the same pass with a check scan. Set Data to CS_ENABLE_READ(1) to enable read with scan, CS_DISABLE_READ(0) to disable. To request the current value set Data to GET_VALUE(-1), which will return the current value in Data. The default is 0 (disabled). A return code of OPOS_E_ILLEGAL is returned if the value is not valid. A return code of OPOS_E_FAILURE is returned if the SureMark Printer not equipped with a document scanner.

Compression format

Table 173. Compression format

Parameter	Type	Description
Command	Long	CS_COMPRESSION_FORMAT (202)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

This command is used to set the compression format when an image is stored or retrieved. Format of an image once stored does not change. To request the current value, set Data to GET_VALUE(-1); the current value ill be returned in Data. The default is CS_CF_TIFF_COMP(0). A return code of OPOS_E_ILLEGAL is returned if the value is not valid. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with a document scanner. The possible values for Data are:

CS_CF_TIFF_COMP(0)
TIFF-CCIT-Group 4 compression

CS_CF_JPEG(1)
JPEG compression

CS_CF_BMP(2)
BMP (uncompressed)

CS_CF_NONE (3)
No compression (gray scale)

CS_CF_TIFF(4)
TIFF file, no compression (gray scale)

Scan document

Table 174. Scan document

Parameter	Type	Description
Command	Long	CS_SCAN_DOCUMENT (203)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

This command is used to start the scan of the document present in the document insert station. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with document scanner or if scan is not successful. String parameter contains the width and length of the scanned document separated by a comma if the scan is successful.

Store document

Table 175. Store document

Parameter	Type	Description
Command	Long	CS_STORE_DOCUMENT (204)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

This command is used to store all or part of the last document scanned. The document image will be stored in the format specified by compression format. The value in Data sets the area or areas to store. Upon return, Data will contain the first index of the storage area in the printer and String will contain the tag data string and the area dimensions to store. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with document scanner or if store is not successful. Possible Data values:

CS_STORE_ENTIRE(0
: store entire image

CS_STORE_PERSONAL(1)
: store personal check template

CS_STORE_BUSINESS(2)
: store business check template

CS_STORE_UD(3)
: store using user defined values

For String, see “Storage string remarks” on page 197 for additional information.

Retrieve by index

Table 176. Retrieve by index

Parameter	Type	Description
Command	Long	CS_RETRIEVE_BY_INDEX (205)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

This command retrieves the image stored at the location specified in Data and stores the image in String. If Data is zero, then the last scanned image is retrieved. The String data format is set based on BinaryConversion. All data starts with a header. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with a document scanner, or if the retrieve is not successful.

Retrieve by tagname

Table 177. Retrieve by tagname

Parameter	Type	Description
Command	Long	CS_RETRIEVE_BY_TAGNAME (206)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

This command retrieves the image stored with the tagData value specified in String and stores it in String. The String data format is set based on OPOS-specific property. A return code of OPOS_E_ILLEGAL is returned if the string is empty. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with document scanner or if the retrieve is not successful. Image is marked as read after retrieval.

Erase storage

Table 178. Erase storage

Parameter	Type	Description
Command	Long	CS_ERASE_STORAGE (207)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

This command erases all stored images from the printer storage. A return code of OPOS_E_FAILURE is returned if the SureMark Printer is not equipped with document scanner or if erase store is not successful.

Storage string remarks

The String parameter for the CS_STORE_DOCUMENT command contains the area or areas to be stored from the last scanned image, in the format:

x0,y0,dx0,dy0,x1,y1,dx1,dy1,x2,y2,dx2,dy2,x3,y3,dx3,dy3,tagdata

where:

x0, y0 2 bytes each, top-left corner of area to be stored

dx0, dy0
2 bytes each, offset in the x and y direction respectively, to be stored

x1, x2, x3, y1, y2, y3,dx1,dx2,dx3,dy1,dy2,dy3
2 bytes each, corresponding offset in the x and y direction to define size of sub-block to store

tagdata
ASCII string null-terminated character, maximum 100 characters

String is interpreted based on the OPOS-specific property. All offsets are based upon the UPOS MapMode. If MapMode is set to PTR_MM_DOTS, 0.01 inches is used.

When storing an image, you can elect to store the entire image within the defined area, or just selected blocks within the defined area. Saving selected blocks reduces storage use, but keeps related data together. When storing multiple blocks of data, each block gets a unique storage location. A byte in the header message indicates that the block is part of a group. The first half-byte indicates which block of the group it is, the second half-byte indicates the number of blocks in the group. If the byte returned is X'14', then the block is the first of a group of four. All blocks contain header information, but only the first block in a group contains tag data. The storage method defines how the image should be stored.

CS_STORE_ENTIRE(0): All data defined within x0, y0, dx0, dy0 is stored; x1, x2, x3, y1, y2, y3, dx1, dx2, dx3, and dy1, dy2, dy3 should not be included.

CS_STORE_PERSONAL(1): A predefined template, based on a personal check is used to store the image. The area defined is the entire personal check, however only the name/address and the MICR blocks are saved; x0, x1, x2, x3, y0, y1, y2, y3, dx0, dx1, dx2, dx3, and dy0, dy1, dy2, dy3 should not be included.

CS_STORE_BUSINESS(2): A predefined template, based on a business check is used to store the image. The area defined is the entire business check, however only the name/address, and MICR blocks are saved; x0, x1, x2, x3, y0, y1, y2, y3, dx0, dx1, dx2, dx3, and dy0, dy1, dy2, dy3 should not be included.

CS_STORE_UD(3): This method allows the user to define sub-blocks of data to store. All command data (x0, x1, x2, x3, y0, y1, y2, y3, dx0, dx1, dx2, dx3, and dy0, dy1, dy2, dy3) must be sent. If fewer than four blocks of data are to be saved, the value of -1 must be sent for unused parameters.

The top-left corner of the scanned document is used as the point of reference (see Figure 32 on page 199). The xn command parameters are horizontal offsets from the left side of the scanned document. The yn parameters are vertical offsets from the top of the scanned document. The dxn and dyn parameters define the width and height, respectively, of the defined area or block.

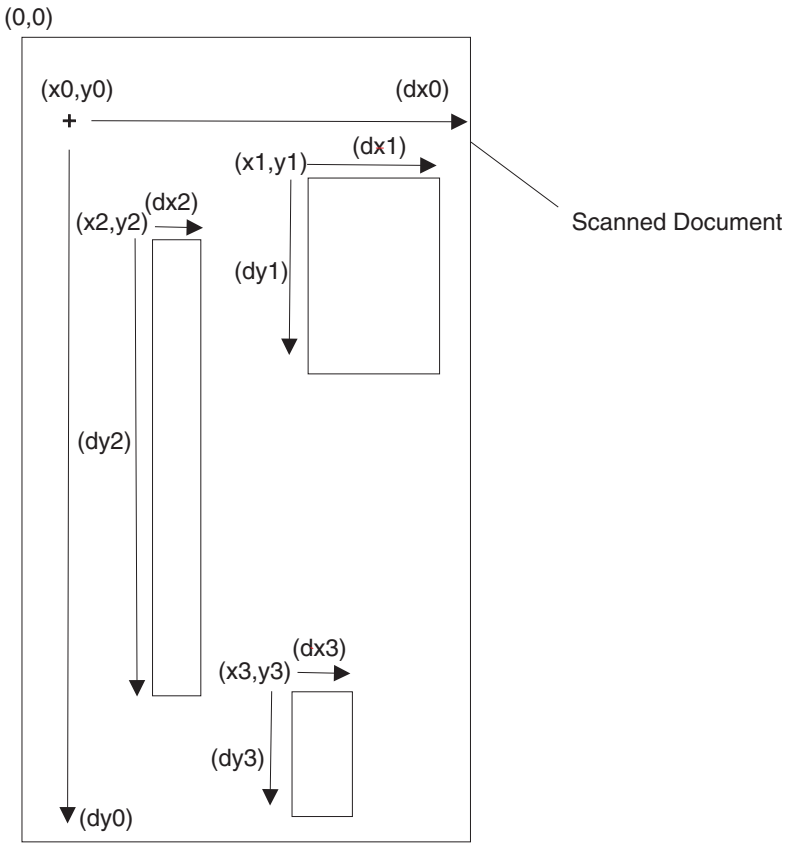


Figure 32. Scanned image organization

Retrieve storage header

Each image retrieved starts with header information in the following format:

- Status: 1 byte
  - Bit 7, Image read; 1 = image has been read, 0 = image has not been read
  - Bits 6 - 0; Reserved
- Compression: 1 byte
  - Bits 7 - 3; Reserved
  - Bits 2 - 0; Algorithm used to compress image. See Compression Format.
- Size1: 4 bytes
  - Number of bytes in image, not including tag data.
- Size2: 4 bytes (x1, y1)
  - Width of image (x1) (in hundredths of inch increments), 2 bytes
  - Height of image (y1) (in hundredths of inch increments), 2 bytes
- Block position: 1 byte
  - Lower four bits indicate the number of blocks that were saved for the image.
  - Upper four bits represent the current block number.
- Future: 1 byte
  - Reserved
- Tag data:
  - ASCII data supplied by the application, terminated by a null string.

Page Mode printing

DirectIO commands are available to support Page Mode printing on the SureMark TI3, TI4, TI8, and TI9 POS Printer. The following table shows the Page Mode supported commands for DirectIO.

Clear print area

Table 179. Clear print area

Parameter	Type	Description
Command	Long	PM_CLEAR_PRINT_AREA (101)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

Clears the print area defined in the current PrintArea.

Page mode print

Table 180. Page mode print

Parameter	Type	Description
Command	Long	PM_PAGE_MODE_PRINT (102)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

Data can be one of the following values:

PM_PM_PAGEMODE (1)
Enters Page Mode

PM_PM_PRINTSAVE (2)
Prints the current page and stays in page mode.

PM_PM_NORMAL (3)
Prints the current page and exits page mode.

PM_PM_CANCEL (4)
Clears the page area and exits page mode.

When the page is printed, the area printed is always from the top of the page area to the end of the print area specified. The top cannot be cropped by using the print area.

Horizontal and vertical position

Table 181. Horizontal and vertical position

Parameter	Type	Description
Command	Long	PM_HORIZONTAL_POSITION (103) PM_VERTICAL_POSITION (104)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

Sets or returns the position for the horizontal/vertical print for the currently selected PageModeStation. If the value sent in position 0 is GETVALUE(-1) the current position is returned. Anything other than GETVALUE attempts to set the position; the value that is set is returned in position 0. Horizontal and vertical position is set based on the print direction. Horizontal position is parallel to the print direction. Vertical position is perpendicular to the print direction.

Page area

Table 182. Page area

Parameter	Type	Description
Command	Long	PM_PAGE_AREA (105)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

Returns the Page area of the printer in the object for the currently selected PageModeStation. The format is “x,y”, where x is the width (horizontal) and y is the length (vertical).

Page mode station

Table 183. Page mode station

Parameter	Type	Description
Command	Long	PM_PAGE_MODE_STATION (106)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

Sets or returns the current PageMode station using position 0 of the data int array. If the value sent in position 0 is GETVALUE(-1) the current station is returned. Anything other than GETVALUE attempts to set the station; the value that is set is returned in position 0.

Print area

Table 184. Print area

Parameter	Type	Description
Command	Long	PM_PRINT_AREA (107)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	String

Remarks:

Sets or returns the desired Print Area. If an empty StringBuffer is sent then a "get only" is emulated. Anything else attempts a set for the currently selected PageModeStation. The format is:

```
x,y,dx,dy
```

where:

- x is the horizontal start position.
- y is the vertical start position.
- dx is the horizontal offset.
- dy is the vertical offset.

Print direction

Table 185. Print direction

Parameter	Type	Description
Command	Long	PM_PRINT_DIRECTION (108)
Data	Long*	Refer to the 'Remarks' section below
Obj	BSTR*	n/a

Remarks:

You must use the Page Mode commands PM_HORIZONTAL_POSITION and PM_VERTICAL_POSITION to align Bar Codes. The printBarCode method alignment parameters are ignored for bar codes printed in Page Mode.

DirectIO commands are also used to set/get properties and capabilities. All commands are indirectly get commands, for example if the HORIZONTALPOSITION is set to a value that is not supported, the data parameter is set to a value that is supported and the hardware is set to that value.

OPOS DirectIO events

NVRAM data

This is received by the application in response to a READ_FLASH_MEMORY DirectIO call.

Table 186. NVRAM data

Parameter	Type	Description
Event Number	Int32	253
Data	Long*	0
Obj	BSTR*	BSTR containing the NVRAM data

NVRAM error

This is received by the application in response to a WRITE_FLASH_MEMORY or READ_FLASH_MEMORY DirectIO call.

Table 187. NVRAM error

Parameter	Type	Description
Event Number	Int32	254

Table 187. NVRAM error (continued)

Parameter	Type	Description
Data	Long*	One of the following values: 0x2A Data received is shorter than the record length. 0x2B NVRAM is full. 0x2C Unable to write record to NVRAM. 0x2D The record number is greater than the maximum number 0x2E The command took more than 5 seconds to complete. 0x2F The record number is greater than the maximum number
Obj	BSTR*	Empty BSTR

OPOS extended result codes

Table 188. Extended result codes

ExtendedResultCode	Name	Description
501	IBM_EPTR_IMAGE_SCAN_FAILED	Image failed to be scanned by the printer.
502	IBM_EPTR_IMAGE_RETRIEVE_FAILED	An error occurred while reading check image data from printer
503	IBM_EPTR_HOME_ERROR	The DI station print head is not in the home position. If opening/closing the DI station cover does not resolve the issue, call IBM and place a hardware service call.
504	IBM_EPTR_PAPER_LOAD_ERROR	A required document was not in the slip station when data was to be printed.
505	IBM_EPTR_PAPER_ERROR	A paper feed error occurred .
506	IBM_EPTR_UNRECOVERABLE_ERROR	Possible at startup if a firmware update corrupted the printer and the printer is running out of the boot sector. Should not happen on TI8/9 which does not update the firmware until after all the update data has been received.
507	IBM_EPTR_COMMAND_REJECT	The printer has rejected a command sent to it. The only time this is likely to occur is if the printer reset in the middle of sending logo data to the printer. When the printer is recovered and the first command is a continue data command containing the rest of the logo data, it will reject the data because it has not been setup to accept the starting logo data.
508	IBM_EPTR_COMMUNICATION_ERROR	Low-level RS-485 communication protocol error.
509	IBM_EPTR_NOT_RESPONDING	An error occurred communicating with the printer.

Table 188. Extended result codes (continued)

ExtendedResultCode	Name	Description
510	IBM_EPTR_RESCAN_REQUIRED	An error occurred scanning the document, and it must be scanned again. Should only occur on TI9 printers, and only if the firmware level is less than ec69.
511	IBM_EPTR_PRINTER_RESET	The printer has reset itself. This is normally caused by the OS re-enumerating the USB bus and causing the printer to power cycle.
512	M_EPTR_NO_IMAGE_AVAILABLE	Check image failed to be read using RetriveByIndex or RetrieveByTagName DirectIO commands.

Additional OPOS information

4610 printer firmware update

The IBM OPOS driver has the ability to automatically update the printer firmware if the printer is attached to the terminal when the terminal boots. It does not have this ability if the printer is attached to the terminal after the terminal has booted and the operating system has loaded.

In addition, printer firmware can be update manually at any time.

Updated printer firmware update files can be downloaded from the IBM website at: http://www.ibm.com/solutions/retail/store/support/

1. Go to: http://www.ibm.com/solutions/retail/store/support
   a. Under Peripherals, click **IBM SureMark Printer**.
   b. Under Downloads, select the appropriate link for your printer model.
   c. From the Search Results displayed, select **IBM 4610 SureMark Printer, Microcode Files for Firmware Update**.
   d. Download the zip file for `OPOS/JavaPOS`. This zip files contains multiple firmware files; choose the correct one from the following for your printer type.
2. Place the firmware file for your printer type into the `\pos\firmware` folder.
3. Make sure the printer is attached and restart.

Each model of 4610 printer has a different firmware file associated with it.

Table 189. IBM 4610 printer firmware update

Printer Model	Firmware update filename
TI1, TI2	aip46mc.hex
TI3, TI4, TG3, TG4, TF6, TM7	aip46mch.hex
TI5, TM7, TF7	aip46mcd.hex
TI8	aip46ti8.hex
TI9	aip46ti8.hex
2CR, 2NR	aip46v4.hex

If the firmware file you select does not match the type of printer you have attached, the program will not update the printer.

Manual update (EIA-232)

Printer firmware in an EIA-232 attached 4610 printer can be updated manually by using **\pos\bin\s232firmwareupdater.exe**.

This is a GUI application that allows you to select a previously-configured printer to update, and which firmware file to use.

Manual update (RS485/USB)

Printer firmware in RS485 and USB attached 4610 printers can be updated manually by using `\pos\bin\aipfld46.exe`.

This is a Windows command-line program that must be run from a Windows shell prompt. Use the following syntax to run the AIPFLD46 program:

```
aipfld46 -Sslot -Pport –Ffirmwarefilename [-O]
```

Table 190. Parameter setting for AIPFLD46

Parameter	Description
-S slot	The slot number of the 4610 for which the firmware download is intended. RS485 is slot number 1, USB is slot number 8.
-P port	The port number of the 4610 for which the firmware download is intended. The port number for RS485 and USB is 17.
-F firmwarefilename	The name of the file containing the firmware data for the printer. A fully qualified path name must be specified when this program is run manually.
-O	By default you can only upgrade firmware in a printer; you can force the application to downgrade printer firmware by using this option.

A log file named `aipfld46.log` is created in the `\pos\log` folder when the firmware download program is executed.

There is no progress indicator on screen, but you can determine the update progress by looking at the log file.

Automatic update (EIA-232)

Because EIA-232 printers can not be automatically detected by the driver, the firmware update program will try to apply all of the printer firmware files to all of the currently configured EIA-232 4610 printers.

The automatic firmware update process is started by a Windows service, 'IBM OPOS RS232 Firmware Update".

If you do not want the firmware update process to run when the terminal boots, you may either disable the service or delete the firmware update files from the `\pos\firmware` folder

Automatic update (RS485/USB)

The automatic firmware update process for RS485 and USB printers is exactly the same as the manual process, except that the `aipfld46` program is run automatically by the driver if it detects that the firmware in the update file is newer than the firmware in the printer.

If you do not want the firmware update process to run when the terminal boots, you should delete the firmware update files from the `\pos\firmware` folder.

4610 printer font conversion

Convert the downloaded files using the 4689CNVT.EXE program.

```
4689CNVT configuration-file
```

configuration-file
One of the configuration files contained in CONVERT.EXE

Notes:

1. The font files must be in the same directory as the conversion program.
2. The output font file will have the same name as the supplied configuration file, but will have a file extension of .FON.

4610 printer font download

To download fonts to the IBM SureMark Point of Sale Printer Model TI5, TM7, and TF7, you will need the following:

- The 4610 SureMark Point of Sale Printer font download program, AIPFNT46.EXE, which comes with the IBM Point of Sale Subsystem for Windows.
- Font files for the IBM 4610 SureMark Point of Sale Printer Model TI5 and the IBM 4610 SureMark Point of Sale Printer Model TI5 font conversion program, 4610CNVT.EXE, which can be downloaded from the IBM Retail Store Solutions Support site: http://www.pc.ibm.com/store
  1. Select **Support** under Retail Store Solutions
  2. Select **IBM SureMark Printer** under SurePOS Peripherals.
  3. Select **DBCS Models** under Downloads. From this page, you can download fonts for Japanese (Gothic and Micho styles plus user-defined fonts), Korean, and Chinese (Simplified and Traditional).
  4. Select **Driver Font Utility Package** to download the font conversion program and font configuration files.

This package contains the following:

File Name	File Description
4610CNVT.EXE	Font conversion program
chnm1381.cfg	Chinese Simplified (PRC) font - Code Page 1381
chnm950b.cfg	Traditional Chinese (BIG-5) with Bold SBCS impact (Taiwan) fonts - Code Page 950
jpngs932.cfg	Japanese Gothic Style font - Code Page 932
jpnms932.cfg	Japanese Mincho Style font - Code Page 932
korms949.cfg	Korean font - Code Page 949

Converting Printer Font Files

Convert the downloaded files using the 4610CNVT.EXE program.

```
4610CNVT configuration-file
```

configuration-file
One of the configuration files contained in CONVERT.EXE.

Notes:

1. The font files must be in the same directory as the conversion program.
2. The output font file will have the same name as the supplied configuration file, but will have a file extension of .FON .

Downloading fonts to the printer

Font files for the IBM 4610 SureMark Point of Sale Printer Model TI5 must be downloaded manually using the AIPFNT46.EXE program:

```
AIPFNT46 -Sslot -Pport -Fx:\dir\font-file {-Q}
```

-Sslot The slot number (in decimal) of the printer for which the download is intended

-Pport The port number (in decimal) of the printer for which the download is intended.

-Fx:\dir\font-file
Fully qualified path name for the font file to be downloaded to the printer.

-Q : Quiet mode. No progress information is displayed.

4689 printer font download

To download fonts to the IBM 4689 SurePOS Receipt Journal Printer, you will need the following:

- The 4689 Printer font download program, AIPFNTRJ.EXE, which comes with the IBM Point of Sale Subsystem for Windows.
- The 4689 Printer font conversion program, 4689CNVT.EXE, and IBM 4689 Printer font files, which can be downloaded from the IBM Retail Store Solutions Support site: http://www.pc.ibm.com/store:
  1. Select **Support** under Retail Store Solutions.
  2. Select **Other Systems and Adapters** under SurePOS Peripherals.
  3. Select **4689** under IBM POS Printers.
  4. Select **Font Support Files** to download the font files.

The download contains the following:

File Name	File Description
4689JPNM.EXE	Japan, Mincho style font (diskette image)
4689JPNG.EXE	Korea, Mincho style font (diskette image)
4689JPNU.EXE	Japan, user-defined fonts (diskette image)
CONVERT.EXE	Contains: • 4689CNVT.EXE : conversion program • jpngs932.cfg: Japan, gothic style + user-defined fonts font configuration file • jpnms932.cfg: Japan, mincho style + user-defined fonts font configuration file • readme.txt: the directions given here

File Name	File Description
readme.txt	The information presented in this section.

Converting Printer Font Files

Convert the downloaded files using the 4689CNVT.EXE program.

```
4689CNVT configuration-file
```

configuration-file
One of the configuration files contained in CONVERT.EXE

Notes:

1. The font files must be in the same directory as the conversion program.
2. The output font file will have the same name as the supplied configuration file, but will have a file extension of .FON.

Downloading Fonts to the Printer

Font files for the IBM 4689 SurePOS Receipt Journal Printer must be downloaded manually using the AIPFNTRJ.EXE program:

```
AIPFNTRJ -Sslot -Pport -Fx:\dir\font-file {-Q}
```

-Sslot :
The slot number (in decimal) of the printer for which the download is intended.

-Pport :
The port number (in decimal) of the printer for which the download is intended.

-Fx:\dir\font-file :
Fully qualified path name for the font file to be downloaded to the printer.

-Q : Quiet mode. No progress information is displayed.

Model 4 printer font download

When the IBM JavaPOS system recognizes a Model 4A printer that it has not used before, it will automatically download to the printer one of the character sets described below:

File	Description
M4A00850.FON	The single-byte characters are the same as those in the Model 4 printer. The characters in this font file are used for the following code pages: • 437 • 850 • 852 • 857 • 860 • 861 • 862 • 863 • 864 • 865
M4A00932.FON	The double-byte characters for Japan. This character set is a subset of the characters in code

	page 932. For the supported characters, see the *IBM JavaPOS system: Installation, Keyboards, and Code Pages* publication.
M4A00949.FON	The double-byte characters for Korea. This character set is a subset of the characters in code page 949. For the supported characters, see the *IBM JavaPOS system: Installation, Keyboards, and Code Pages* publication.

Manually Downloading Characters

If at a later time, you want to use another character set, or if you want to define your own characters and download them to the Model 4A printer, you can do so manually. The program to download a new character set to the Model 4A printer is AIPM4ALD.EXE. This program is in the `\BIN` directory where the IBM JavaPOS system is installed.

Use the following syntax to run the AIPM4ALD program:

```
AIPM4ALD -Sslot -Pport -Fx:\dir\font-file [ -Q ]
```

The parameters for the AIPM4ALD program are:

Parameter	Description
-S *slot*	The slot number (in decimal) of the IBM Model 4A printer to which the new character set is to be downloaded.
-P *port*	The port number (in decimal) of the IBM Model 4A printer to which the new character set is to be downloaded.
-F *x:\dir\font-file*	The name of the file containing the characters to be downloaded to the IBM Model 4A printer. A fully-qualified path name must be specified when this program is run manually.
-Q	Quiet Mode. Do not display progress information
-V	Verify Characters. Verify each character by printing it on the CR station after all the characters have been downloaded.

Font File Format

There are three types of records that are allowed in a Model 4A font file. These are:
- Comment Record
- Keyword Record
- Character Definition Record

A *comment record* can be either a blank line, or a line with an exclamation point (!) as the first non-blank character.

A *keyword record* is a line with one of the keywords defined below followed by one or more blank characters and ending with a value for the keyword.

A *character definition record* is a line of hexadecimal ASCII characters. Each pair of characters defines one byte of the character definition. The first two pairs of characters identify the character being defined and the remaining characters define the character.

Keywords: To define characters to the IBM JavaPOS system for use with the Model 4A printer, the following keywords and their corresponding values must be defined in the font file along with the definition of each character. If any of these keywords are omitted, unpredictable results can occur when your application is using the printer.

Keyword	Description
CODEPAGE	The code page this font file defines.
HEIGHT	Defines the double-byte character height. The valid values are 9 or 16.
MODE	Defines this code page to be a single-byte, or a double-byte code page. A value of 0 (zero) indicates this is a single-byte code page and a value of 1 (one) indicates this is a double-byte code page.
SPACE	The number of dot columns to be added to each double-byte character. The valid values are 0 (zero) through 16.
WIDTH	Defines the double-byte character width. The valid values are 10 through 16.

Character Definition Record: Each *character definition record* defines one character for the Model 4A printer. Each character on the line must be a hexadecimal ASCII character. Each pair of characters defines one byte of the character definition. The first two pairs of characters identify the character being defined and the remaining characters define the character.

Character Identifier: The first two pairs of characters that identify the character being defined must be between 0x0000 and 0x00FF, or between 0x8000 and 0xFFFF. The characters in the range 0x0000 through 0x00FF define the single-byte characters and there must be 44 hexadecimal ASCII characters (22 bytes). The characters in the range 0x8000 through 0xFFFF define the double-byte characters and the number of hexadecimal ASCII characters depends upon the width of the character being defined. The number of characters in the line can be calculated by multiplying the character width, as defined by the WIDTH keyword, by 4 and adding 4. The additional four characters are for the character identifier.

Note: If you define double-byte characters in your font file, you must define the character 0x8000 to be all zeros. This is because the IBM JavaPOS system uses that character for padding double-byte character print lines when positioning the print head over characters already printed.

Character Definition: Each dot column on the Model 4A printer has nine print wires for single-byte characters and between nine to sixteen print wires for double-byte characters.

The definition of each character follows the character identifier on each line. Each pair of hexadecimal ASCII values defines one byte of the character. For the Model 4A printer, each pair of bytes (four characters) defines which print wires are turned on for each dot column to be printed. Within each pair, the first two hexadecimal ASCII characters (first byte) defines which of the lower eight print wires will be turned on and the second two hexadecimal ASCII characters (second byte) defines which of the upper eight print wires will be turned on. Within each byte, the least significant bit represents the top print wire.

For single-byte characters, the least significant bit of the first byte in the pair of dot column bytes represents print wire number nine. For single-byte characters, this is

the only bit used in the first byte. For double-byte characters, the number of significant bits is determined by the LENGTH keyword. If the LENGTH keyword is set to 9, the double-byte character definition is the same as the single-byte character. If the LENGTH keyword is set to 16, then all eight bits of the first byte is used to define the print wires 9 through 16.

Character Definition Restrictions: The following should be considered when defining characters for the Model 4A printer:

- When defining characters, the same print wire should not be turned on in consecutive dot columns.
- For single-byte characters, the spacing between characters is included in the character definition.
- For double-byte characters, the spacing between characters can be either included in the character definition, or defined by the SPACE keyword.

Barcode printing

The width parameter of the printBarcode method is not a definite value. Since the ratio of the width of thick and thin lines must be fixed, the width parameter is used to calculate a percentage of the total line width of the station, such as RecLineWidth. The printer hardware accepts values of 2 - 4. Therefore, if width is less than 34% of XxxLineWidth, then 2 is used, 34% to 66% sends a 3 and greater than 66% sends a 4.

The height parameter for the Receipt Station accepts 1 - 255 dot rows. On the slip station, it height parameter is converted to head passes. The acceptable number of head passes is 3 to 5, or 27 to 45 dots in map mode, when SlpLineSpacing equals 9 dots.

Rotated printing

On the slip, only the alignment settings are ignored. Bold and Double High attributes are ignored by the hardware.

DBCS support

Downloading of fonts to SureMark printers that support DBCS is described in *Point of Sale Subsystem:Programming Reference and User's Guide, Appendix F*.

Color printing

In order to use color printing capability, the firmware level must be 33 or greater, supported color thermal paper must be used, and the CapRec2Color option must be set at configuration time.

User-defined fonts

The SureMark printers support up to four user-defined character sets. For EIA-232 printers, the user-defined character sets are identified as character sets 102 through 105; for RS-485 and USB printers, the user-defined character sets are identified as 101 through 104. These character set identifiers appear in the CharacterSetList property.

Note: For EIA-232 printers, character set 101 is the printer's generic code page. For RS-485 and USB, the printer's generic code page is not supported.

To define a single proportional font, two of the printer's user-defined character sets are required. For EIA-232, a user-defined proportional font can be either character set 102 or 104; for RS-485 and USB, a user-defined proportional font can be either character set 101 or 103.

Only two user-defined fonts are supported on the impact station. When using a printer that has an impact station:

1. The odd-numbered user-defined character sets on the receipt station map to user-defined character set 1 on the slip station; even numbered character sets on the receipt station map to slip user-defined character set 2.
2. If the application uses user-defined character sets with a printer that has both receipt and slip stations, then both receipt and slip character sets should be downloaded to the printer.

Proportional font support

The printer drivers default to fixed width. In order to switch between fixed and proportional fonts, the Font Typeface is used. To illustrate this difference, the FontTypefaceList displays values "Fixed, Proportional" only when the CharacterSet has been set to a Proportional User Defined Font (101 or 103 for RS-485/USB, or 102 and 104 for EIA-232). This is only valid for User Defined Proportional Fonts. The resident fonts are fixed width.

The OPOS Specification assumes that all characters are fixed width. Proportional font printing is supported, providing the following:

1. If a proportional font is active and the current font typeface is set to proportional, all properties such as RecLineChars, RecLineHeight and RecLineSpacing are set to zero and RecCharList is set to a null string. The downloaded font determines the properties of the printed line.

   Based on this assumption, we do not wrap lines in proportional mode when the number of characters on a line is greater than RecLineChars. Instead, it is up to the application developer to send a line feed when the print line is complete or the printer feeds when it has reached the end of the line.
2. Text alignment is based on the fact that a fixed number of characters fit on a line. This is not possible with proportional fonts. Therefore, when the printer is in proportional mode, instead of formatting the line within the SO, we pass on the Alignment Escape Sequences within the printer to let the printer format the alignment. This allows the printer to center text or split left and right aligned text at the hardware.

   Finally, if the font typeface is set to Fixed when a proportional font is used, the printer prints the characters a fixed distance apart, and all of the Line properties are valid. The actual fixed width is set in the registry entry for the printer, using the keyword `"ProportionalFontFixedWidth"`. The range of values the printer supports is 8 to 32. If this value is not specified, the control uses a width of half of the height and then adjust it for a best fit within the valid range.

Tone device sharing

Certain models for SureMark printers are equipped with tone devices. While the OPOS model states that tone indicators are shareable devices, this device cannot be shared from separate applications. Trying to enable this device from two applications will cause the enable to fail for the second application. If using the tone indicator from two applications, it is recommended that each application claim the device before using it.

Code 128 A/B/C support

The Code 128 Bar Code Symbology has three code sets and also includes some special characters that indicate a change in code set, a function, or a shift from Code Set A to B, or vice versa. Table 191 lists the characters for each code set.

Table 191. Code 128 character sets

Code set	Character set
Code A	X'00'-X'5F', FNC1, FNC2, FNC3, FNC4, SHIFT, CODE B, CODE C
Code B	X'20'-X'7f', FNC1, FNC2, FNC3, FNC4, SHIFT, CODEA, CODE C
Code C	X'00'-X'63' for decimal values 00-99, FNC1, CODE A, CODE B

Characters are mapped from ASCII to the corresponding value for the selected code set. In Code Sets A and B, this is a one-to-one mapping. In Code Set C, each two digits is converted to a single value. A sentinel character, the left curly bracket ({)followed by a particular character, is used to indicate a special character. A starting code set is required at the start of the data. The symbology value to use is 123.

Table 192 lists the character pairs for encoding the special characters:

Table 192. Code 128 special character encoding

Special character	Character pair for encoding
SHIFT	{S
CODE A	{A
CODE B	{B
CODE C	{C
FNC1	{1
FNC2	{2
FNC3	{3
FNC4	{4
{	{{

SureOne/SurePOS 100 DBCS printer limitations

For a DBCS printer, the printer code page is assumed to be the same as the locale of Windows.

Improving printer performance

When AsyncMode is *false*, IBM's OPOS Printer drivers verify that each print line is printed on the paper in synchronous mode, not just that it has been sent to the printer. This approach provides the application with an accurate completion status, but as a result, the drivers appear to be slower than the rated printer throughput. To improve printer driver throughput, do the following:

- Set AsyncMode to *true* so that groups of printer lines are sent out to the printer as they are received by the driver.
- Use Transaction Mode.
- Group lines in single print command.
- Format a full line.

Chapter 17. Scanner (bar code reader)

Supported devices

Table 193. Scanner supported devices

Device	Connectivity
1. IBM 1520 Compatible Hand-held Scanner	RS-485
2. IBM 4687 Compatible Scanner	RS-485
	USB (using Protocol Converter)
3. IBM 4696 Compatible Scanner	RS-485
	USB (using Protocol Converter)
4. IBM 4697 Compatible Scanner	RS-485
	USB (using Protocol Converter)
5. IBM 4698 Compatible Scanner	RS-485
	USB (using Protocol Converter)
6. IBM HHBCR Compatible	RS-485
7. IBM HHBCR2 Compatible	RS-485
8. OEM Scanner	USB (must conform to *IBM USB OEM Point-of-Sale Device Interface Specification* V1.0 or later)
9. IBM 4685 Scanner	RS-485, USB
10. Kiosk scanner	Integrated

Supported properties and methods

Table 194. Scanner common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	All	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	All	
DataEventEnabled	All	
DeviceControlDescription	All	
DeviceControlVersion	All	

Table 194. Scanner common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 195. Scanner specific properties

Property	JavaPOS and OPOS Gateway	OPOS
DecodeData	All	
ScanData	All	
ScanDataLabel	All	
ScanDataType	All	

Table 196. Scanner common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	All	
clearOutput	All	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 197. Scanner events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	
DirectIOEvent	Not supported	
ErrorEvent	All	
StatusUpdateEvent	All	

JavaPOS configuration

This section contains device-specific notes for the EIA-232, RS-485 and USB scanners. Scanner configuration is done through the jpos.xml file. An example of a typical configuration property is:

```
<prop name="setEnableCODE39" type="Boolean" value="true"/>
```

Notes:

1. Each scanner's hardware capabilities determine which symbologies it can decode. Enabling a symbology property does not guarantee that the attached scanner can decode that symbology. See the Web site of the scanner vendor for more information about the supported symbologies and hardware capabilities.
2. When RS-485 scanners are attached to the system using a USB protocol converter, the supported symbologies are unchanged.
3. To read supplemental data on a 4685 scanner, at least one of the labels must be disabled. For example, setXXX label property to false in JposEntry for this scanner. The only way to enable all supported labels is to set the scanner to ″test mode″. In test mode, the 4685 reads all the supported labels, except for supplementals. To read supplementals, at least one label must be disabled.
4. If the device JposEntry in the jpos.xml does not contain a specific property, that property will be initialized to the default value specified in this document. Otherwise the value specified for that property in the JposEntry will be used.

Table 198 lists the supported symbologies for IBM scanners.

Table 198. Supported symbologies for IBM scanners

Symbology	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
UPC-A	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
UPC-E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
UPC D1					✓	✓	✓	✓		
UPC D2					✓	✓	✓	✓		
UPC D3			✓	✓	✓	✓	✓	✓		
UPC D4					✓	✓	✓	✓		
UPC D5					✓	✓	✓	✓		
EAN/JAN 8		✓	✓	✓	✓	✓	✓	✓	✓	✓
EAN/JAN 13		✓	✓	✓	✓	✓	✓	✓	✓	✓
ITF	✓	✓	✓	✓		✓	✓	✓	✓	✓
CODABAR			✓	✓		✓		✓	✓	✓

Table 198. Supported symbologies for IBM scanners (continued)

Symbology	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
CODE 39	✓	✓	✓	✓		✓	✓	✓	✓	✓
CODE 93			✓	✓		✓		✓	✓	✓
CODE 128			✓	✓		✓	✓	✓	✓	✓
Standard 2 of 5									✓	✓
UCC/EAN128								✓	✓	✓
GS1 DataBar Omnidirectional									✓	✓
GS1 DataBar Expanded									✓	✓
PDF-417										✓
Maxicode OCR-A OCR-B								✓		

Table 199 lists the configurable parameters for scanners and indicates which parameters are supported by each scanner model.

Table 199. Scanner configurable parameters

Parameter	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
“enableLaserOnOffSwitch” on page 220								✓		
“enableProgrammingViaBarcodes” on page 220					✓	✓	✓	✓		✓
“enableSwitchControlledVolumeAdjust” on page 220					✓	✓	✓	✓		
“setBarCodes1” on page 221	✓	✓	✓	✓	✓	✓	✓	✓		
“setBarCodes2” on page 222	✓			✓		✓	✓	✓		
“setBarCodes3” on page 222	✓						✓	✓		
“setBarCodes4” on page 222	✓						✓	✓		
“setBeeperDuration” on page 222					✓	✓	✓	✓	✓	
“setBeeperFrequency” on page 223					✓	✓		✓	✓	✓
“setBeeperVolume” on page 223					✓	✓	✓	✓		
“setCheckModulo” on page 223		✓	✓	✓						
“setCode128ScansPerRead” on page 223							✓			
“setCode39ScansPerRead” on page 224							✓			
“setDecodeAlgorithm” on page 224					✓	✓	✓			
“setDoubleReadTimeOut” on page 224					✓	✓	✓	✓		
“setDTouchMode” on page 225		✓	✓	✓						
“setEAN13ScansPerRead” on page 225					✓	✓	✓			
“setEAN8ScansPerRead” on page 226					✓	✓	✓			
“setEnable_2_DigitSupplementals” on page 226				✓			✓	✓	✓	
“setEnable_5_DigitSupplementals” on page 226				✓			✓	✓	✓	
“setEnableCodabar” on page 226			✓	✓		✓		✓	✓	✓
“setEnableCode128” on page 227			✓	✓		✓	✓	✓	✓	✓
“setEnableCode128Supplementals” on page 227								✓		
“setEnableCODE39” on page 227	✓	✓	✓	✓		✓	✓	✓	✓	✓
“setEnableCode39CheckDigit” on page 227								✓		
“setEnableCode93” on page 227			✓	✓		✓		✓	✓	✓
“setEnableEAN_JAN_TwoLabelDecoding” on page 227						✓	✓	✓		

Table 199. Scanner configurable parameters (continued)

Parameter	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
“setEnableGoodReadBeep” on page 228	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
“setEnableInterleaved2of5” on page 229	✓	✓	✓	✓		✓	✓	✓	✓	✓
“setEnableITFCheckDigit” on page 229								✓		
“setEnablePDF417” on page 229										✓
“setEnableRSS14 (deprecated)” on page 229									✓	✓
“setEnableRSS_Expanded (deprecated)” on page 230									✓	✓
“setEnableStandard2of5” on page 230									✓	✓
“setEnableUCC_EAN128” on page 230								✓	✓	✓
“setEnableUPC_A_CheckDigit” on page 230								✓	✓	✓
“setEnableUPC_A_To_EAN13Expansion” on page 231					✓	✓	✓	✓	✓	
“setEnableUPC_E_CheckDigit” on page 231								✓	✓	✓
“setEnableUPC_E_To_EAN13Expansion” on page 231					✓	✓	✓	✓		
“setEnableUPC_E_To_UPC_AExpansion” on page 231							✓	✓		✓
“setEnableUPCAE_EANJAN813” on page 232	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
“setEnableUPCD1D5” on page 232			✓		✓	✓	✓	✓		
“setEnableVerificationUPC_A_EAN13_fiveDigit” on page 232					✓	✓	✓	✓		
“setEnableVerificationUPC_A_EAN13_fourDigit” on page 232					✓	✓	✓	✓		
“setITFLength1” on page 232	✓					✓	✓	✓	✓	✓
“setITFLength2” on page 233						✓	✓	✓	✓	✓
“setITFLengths” on page 233								✓	✓	✓
“setITFLengthSpecifiedTwo” on page 233						✓	✓	✓		
“setITFScansPerRead” on page 234							✓			
“setLaserTimeOut” on page 234					✓	✓	✓	✓		
“setLED_GoodRead_Duration” on page 234								✓		
“setMotorTimeOut” on page 235					✓	✓	✓	✓		
“setSecurityLevelForInStore” on page 235								✓		
“setSTFLength1” on page 236									✓	✓
“setSTFLength2” on page 236									✓	✓
“setSTFLengths” on page 236									✓	✓
“setStoreScansPerRead” on page 236					✓	✓	✓			
“setSupplementals” on page 237								✓		✓
“setSupplementalsSecurityLevel” on page 237									✓	✓
“setUPCAScansPerRead” on page 237					✓	✓	✓			
“setUPCDScansPerRead” on page 238					✓	✓	✓			
“setUPCEScansPerRead” on page 238					✓	✓	✓			

enableLaserOnOffSwitch

Property Type: boolean

Default: false

Models Supported: USB OEM

This property controls whether the laser power switch on the scanner unit is enabled or disabled. When a scanner is on, its motor is running and its laser is active. When a scanner is off, its motor is not running and its laser is inactive. When the laser power switch is enabled, it can be used to turn the scanner off. When the laser power switch is disabled, it cannot be used to turn the scanner off. The laser power switch can always be used to turn the scanner on.

enableProgrammingViaBarcodes

```
<prop name="enableProgrammingViaBarcodes" type="Boolean" value="false" />
```

This property controls whether or not the scanner can be programmed using the manufacturer-supplied programming bar codes.

Supported on: 4696, 4697, 4698, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner.

enableSwitchControlledVolumeAdjust

Property Type: boolean

Default: false

Models Supported: 4696,4697,4698 and USB OEM

This property enables and disables switch-controlled beep volume adjustment.

setBarCodes1

Property Type: byte

Default: 00

Models Supported: 1520, 4500, 4501, 4685, 4696, 4697, 4698, USB OEM

This property is one of several properties which determine what combination of bar code types a scanner recognizes. When configured in a specific mode, a scanner recognizes and returns all bar code types that are associated with that mode. Table 200 lists the values that can be given for the setBarCodes1 property for both RS-485 and USB-attached scanners.

Table 200. setBarCodes1 values

Resource Value	Bar Code Types
0 - GROUP_NONE	None
1 - GROUP_UPC_EAN_ITF	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13, Interleaved 2 of 5 (ITF)
2 - GROUP_UPCAED	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-D1, UPC-D2, UPC-D3, UPC-D4, UPC-D5
3 - GROUP_UPC_EAN_CODE128	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13, Code128
4 - GROUP_UPC_EAN_CODE93	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13, Code93
5 - GROUP_UPC_EAN_CODE39	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13, Code39
6 - GROUP_UPC_EAN_CODABAR	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13, Codabar
7 - GROUP_UPC_EAN_2_5_CODABAR	UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13UPC-A, UPC-A+2/+5, UPC-E, UPC-E+2/+5, EAN/JAN-8, EAN/JAN-8+2/+5, EAN/JAN-13, EAN/JAN-13+2/+5,Codabar (See Note.)

Note: Different scanners handle supplementals (+2, +5) differently. When supplementals are enabled, some scanners will only read UPC-A, UPC-E, EAN/JAN-8 and EAN/JAN-13 barcodes with a supplemental; other scanners will read UPC-A, UPC-E, EAN/JAN-8 and EAN/JAN-13 barcodes with and without a supplemetal.

setBarCodes2

Property Type: byte

Default: 0

Scanner models supported: 1520, 4685, 4697, 4698, USB OEM

This property is one of several properties which determine what combination of bar code types a scanner recognizes. When configured in a specific mode, a scanner recognizes and returns all bar code types that are associated with that mode.

The values for the setBarCodes2 property are the same as those for setBarCodes1 property.

setBarCodes3

Property Type: byte

Default: 0

Models Supported: 1520, 4698, USB OEM

This property is one of several properties which determine what combination of bar code types a scanner recognizes. When configured in a specific mode, a scanner recognizes and returns all bar code types that are associated with that mode.

The values for the setBarCodes3 property are the same as those for setBarCodes1 property.

setBarCodes4

Property Type: byte

Default: 0

Models Supported: 1520, 4698, USB OEM

This property is one of several properties which determine what combination of bar code types a scanner recognizes. When configured in a specific mode, a scanner recognizes and returns all bar code types that are associated with that mode.

The values for the setBarCodes4 property are the same as those for setBarCodes1 property.

setBeeperDuration

Property Type: byte

Default: 0

Models Supported: USB OEM , 4696, 4697, 4698, AnyPlace Kiosk Line Scanner

This property specifies the duration of the tone that the beeper makes upon a successful read. This property can have the following values:

0 SHORT: Use the shortest time value for the beeper duration.

1 MEDLONG: Use a medium time value for the beeper duration.

2 LONG: Use a longtime value for the beeper duration.

3 LONGEST: Use the longest time value for the beeper duration.

setBeeperFrequency

Property Type: byte

Default: 0

Models Supported: 4696, 4697, USB OEM, AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

This property specifies the frequency of the tone that the beeper makes upon a successful read.

0 Lowest: The beep of the scanner has a very low frequency.

1 Low: The beep of the scanner has a low frequency.

2 High: The beep of the scanner has a high frequency.

3 Highest: The beep of the scanner has a very high frequency.

setBeeperVolume

Property Type: byte

Default: 0

Models Supported: 4696, 4697, 4698, USB OEM

This property specifies the volume of the tone that beeper makes. This property can have the following values:

0 Lowest: The volume of the beep is set at its lowest level.

1 Low: The volume of the beep is set at a low level.

2 Medium: The volume of the beep is set at a medium level.

3 High: The volume of the beep is set at a high level.

setCheckModulo

Property Type: boolean

Default: false

Models Supported: 4500, 4501, 4685

This property controls whether or not the scanner checks the modulo byte of a bar code to ensure that it is correct before returning the bar code to an application.

setCode128ScansPerRead

Property Type: byte

Default: 1

Models Supported:4698

This property controls the minimum number of scans performed for Code 128 labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the default value should be used.

setCode39ScansPerRead

Property Type: byte

Default: 1

Models Supported:4698

This property controls the minimum number of scans performed for Code 39 labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the default value should be used.

setDecodeAlgorithm

Property Type: byte

Default: 0

Models Supported: 4696, 4697, 4698

This property controls the use of decode algorithms. These decode algorithms use a complex set of tests to assemble bar code data from damaged or truncated labels. These techniques also give faster read performance on good bar codes. This property can have the following values:

0 Label assembly

1 marginless

Note: Label assembly can be selected only if Label assembly was enabled during factory configuration, if Label assembly was not enabled during factory configuration, the selection shall be considered valid but the parameter shall be forced to marginless.

setDoubleReadTimeOut

Property Type: byte

Default: 0

Models Supported: 4696, 4697, 4698, USB OEM

This property controls the length of the double-read timeout. Most scanners decode and recognize a bar code label several times as the bar code is passed through the scanning region. To prevent a scanner from returning data from the same bar code

several times, scanners are programmed with a double-read timeout. The double-read timeout is the length of time that the scanner waits before returning the same bar code data twice.

This property can have the following values for 4696, 4697 and 4698:

0 Use 500 ms for the double-read timeout

70 Use 700 ms for the double-read timeout

90 Use 900 ms for the double-read timeout

This property can have the following values for USB:

X'00' Use the shortest time value for the double-read timeout

X'20' Use a moderate time value for the double-read timeout

X'40' Use the longest time value for the double-read timeout

setDTouchMode

Property Type: boolean

Default: false

Models Supported: 4500, 4501, 4685

This property controls the state of the double-touch mode of the scanner. Double-touch mode allows scanners such as the hand-held bar code reader to read bar codes that are larger than the reading head. In double-touch mode, when the read head of the scanner is placed over the first half of the label, the scanner emits a repetitive beeping noise (if the beeper is enabled) to indicate that the data was read. When the read head is placed over the second half of the label, the data is sent to the system unit. Putting a scanner in double-touch mode does not prevent it from reading a bar code in a single touch. Only bar codes such as UPC-A, EAN-13, and UPC-D3 can be read using double-touch mode. For UPC-D3 bar codes, double-touch mode is automatically enabled by the scanner and cannot be switched off.

setEAN13ScansPerRead

Property Type: byte

Default: 1

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for EAN-13 labels. This is the number of scans for a single pass of an item over the scanner window. The value of this resource has a range from zero to four. Any value outside this range is not valid. A value of zero indicates that the number of scans per read is not specified by the application. In this case, the default value is used.

This property can have the following values:

1 One scan

2 Two scans

3 Three scans

4 Four scans

setEAN8ScansPerRead

Property Type: byte

Default: 2

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for EAN-8 labels.

This resource can have the following values:

1 One scan

2 Two scans

3 Three scans

4 Four scans

setEnable_2_DigitSupplementals

Property Type: boolean

Default: false

Models Supported: 4685, 4698, USB OEM, AnyPlace Kiosk Line scanner

This property controls whether the scanner recognizes 2-digit supplementals for UPC, UPE, EAN/JAN8, and EAN/JAN13. When enabled, supplementals are optional for all these symbologies.

setEnable_5_DigitSupplementals

Property Type: boolean

Default: false

Models Supported: , 4685, 4698, USB OEM, AnyPlace Kiosk Line scanner

This property controls whether the scanner recognizes 5-digit supplementals for UPC-A, UPC-E, EAN/JAN8, and EAN/JAN13. When enabled, supplementals are optional for all these symbologies.

setEnableCodabar

Property Type: boolean

Default: false

Models Supported: 4501, 4685, 4697, USB OEM, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni Scanner

Enable or disable the CodaBar barcodes.

setEnableCode128

Property Type: boolean

Default: false

Models Supported: 4501, 4685, 4697, 4698, USB OEM, AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enable or disable the Code128 barcodes.

setEnableCode128Supplementals

Property Type: boolean

Default: false

Models Supported:USB OEM

Enable or disable supplementals for code128 barcodes.

setEnableCODE39

Property Type: boolean

Default: false

Models Supported: 1520, 4500, 4501, 4685, 4697, 4698, USB OEM, AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enable or disable the Code39 barcodes.

setEnableCode39CheckDigit

Property Type: boolean

Default: false

Models Supported: USB OEM

Enable or disable if check digits should be transmitted for code39 barcodes.

setEnableCode93

Property Type: boolean

Default: false

Models Supported: 4501, 4685, 4697, USB OEM

Enable or disable the Code93 barcodes.

setEnableEAN_JAN_TwoLabelDecoding

Property Type: boolean

Default: false

Models Supported: 4697, 4698, USB OEM

Enable or disable the EAN/JAN Two-label decoding.

setEnableCompositeCC_A

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Omni Scanner.

Enable or disable decoding of CC_A (or CC_B or CC_C) composite barcodes. As composite barcodes are 2D barcodes that appear on top of a 1D barcode (such as UPCA, EAN13, EAN128, and so on), note that the corresponding 1D barcode must be enabled too. Enabling this or any other 2D barcode disables the omnidirectional feature because 2D barcodes cannot be scanned in omnidirectional mode.

setEnableCompositeCC_B

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Omni Scanner.

Enable or disable decoding of CC_A (or CC_B or CC_C) composite barcodes. As composite barcodes are 2D barcodes that appear on top of a 1D barcode (such as UPCA, EAN13, EAN128, and so on), note that the corresponding 1D barcode must be enabled too. Enabling this or any other 2D barcode disables the omnidirectional feature because 2D barcodes cannot be scanned in omnidirectional mode.

setEnableCompositeCC_C

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Omni Scanner.

Enable or disable decoding of CC_A (or CC_B or CC_C) composite barcodes. As composite barcodes are 2D barcodes that appear on top of a 1D barcode (such as UPCA, EAN13, EAN128, and so on), note that the corresponding 1D barcode must be enabled too. Enabling this or any other 2D barcode disables the omnidirectional feature because 2D barcodes cannot be scanned in omnidirectional mode.

setEnableGoodReadBeep

Property Type: boolean

Default: false

Models Supported: All

Enable or disable GoodReadBeep.

setEnableInterleaved2of5

Property Type: boolean

Default: false

Models Supported: 1520, 4500, 4501, 4685, 4697, 4698, USB OEM, AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enable or disable the Interleaved 2-of-5 (ITF) barcodes.

setEnableITFCheckDigit

Property Type: boolean

Default: false

Models Supported: USB OEM

Enable or disable if check digits should be transmitted for Interleaved 2-of-5 (ITF) barcodes.

setEnablePDF417

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Omni Scanner

Enable or disable decoding of PDF417 barcodes. Enabling this or any other 2D barcode disables the omnidirectional feature because 2D barcodes cannot be scanned in omnidirectional mode.

setEnableMicroPDF417

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Omni Scanner

Enable or disable decoding of Micro PDF417 barcodes. Enabling this or any other 2D barcode disables the omnidirectional feature because 2D barcodes cannot be scanned in omnidirectional mode.

setEnableRSS14 (deprecated)

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the Reduced Space Symbology RSS14 barcodes.

setEnableGS1Databar

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the GS1 Databar barcodes.

setEnableRSS_Expanded (deprecated)

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the RSS Expanded barcodes.

setEnableGS1Databar_Expanded

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the GS1 Databar Expanded barcodes.

setEnableStandard2of5

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

Enable or disable the Standard 2-of-5 (STF) barcodes.

setEnableUCC_EAN128

Property Type: boolean

Default: false

Models Supported: USB OEM, AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enable or disable the UCC/EAN128 barcodes.

setEnableUPC_A_CheckDigit

Property Type: boolean

Default: false

Models Supported: USB OEM, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

Enable or disable if check digits should be transmitted for UPC-A barcodes.

setEnableUPC_A_To_EAN13Expansion

Property Type: boolean

Default: false

Models Supported: 4696, 4697, 4698, USB OEM, AnyPlace Kiosk Line scanner

This property controls the report format for UPC-A labels. UPC-A and EAN-13 are part of the same numbering system. It is possible to have the scanner report all of these codes in a uniform format. UPC-A is a 12-digit subset of EAN-13. The scanner can add a leading 0 (zero) to the UPC-A number, yielding its EAN equivalent.

setEnableUPC_E_CheckDigit

Property Type: boolean

Default: false

Models Supported: USB OEM, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

Enable or disable if check digits are transmitted for UPC-E barcodes.

setEnableUPC_E_To_EAN13Expansion

Property Type: boolean

Default: false

Models Supported: 4696, 4697, 4698, USB OEM

This property controls the report format for UPC-E labels. UPC-E and EAN-13 are part of the same numbering system. It is possible to have the scanner report all of these codes in a uniform format. UPC-E is a short form of a UPC-A number, UPC-A is a 12-digit subset of EAN-13. The scanner can add a leading 0 (zero) to the UPC-E number, yielding its EAN equivalent.

setEnableUPC_E_To_UPC_AExpansion

Property Type: boolean

Default: false

Models Supported: 4698, USB OEM, AnyPlace Kiosk Omni Scanner

This controls the report format for UPC-E labels. UPC-A, UPC-E, and EAN-13 are all part of the same numbering system. It is possible to have the scanner report all of these codes in a uniform format. The scanner can expand UPC-E data to its UPC-A format.

setEnableUPCAE_EANJAN813

Property Type: boolean

Default: false

Models Supported: All

Enable or disable the UPC-A, UPC-E, EAN/JAN 8 and EAN/JAN 13 barcodes.

setEnableUPCD1D5

Property Type: boolean

Default: false

Models Supported: 4501, 4696, 4697, 4698, USB OEM

Enable or disable the UPC D1 to UPC D5 barcodes.

setEnableVerificationUPC_A_EAN13_fiveDigit

Property Type: boolean

Default: false

Models Supported: 4696, 4697, 4698, USB OEM

This controls whether the scanner verifies the 5 digit price check character. UPC and EAN specifications allow for a price check character to be included in the digits encoded on in-store random weight items. This property is mutually exclusive with UPC-A, EAN-13 four Digit Verfication.

setEnableVerificationUPC_A_EAN13_fourDigit

Property Type: boolean

Default: false

Models Supported: 4696, 4697, 4698, USB OEM

This property controls whether the scanner verifies the 4 digit price check character. UPC and EAN specifications allow for a price check character to be included in the digits encoded on in-store random weight items. This property is mutually exclusive with UPC-A, EAN-13 five Digit Verfication.

setITFLength1

Property Type: byte

Default: 0

Models Supported: 1520, 4697, 4698, USB OEM. AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

For the scanners that support Interleaved 2-of-5 bar codes, this property can be used to specify one valid length for Interleaved 2-of-5 bar codes. This value indicates the exact length of the Interleaved 2-of-5 bar codes that the scanner

reads. If an Interleaved 2-of-5 bar code is not of the correct length, then the bar code is not read by the scanner. The value of the setITFLength1 property must be an even number from 4 to 32. (For the IBM 1520-A02, the value must be an even number from 4 to 30.) This value specifies the exact length of the bar code.

If the scanner is not configured to read Interleaved 2-of-5 bar codes, the value of this property is ignored. This property is provided because scanners are prone to errors when reading Interleaved 2-of-5 labels. Because these labels are of variable length, it is possible for a scanner to read only part of a label, but process it as though it had read the complete label. If an application is only expecting Interleaved 2-of-5 labels of a certain length, this property ensures that the scanner does not read any partial labels.

setITFLength2

Property Type: byte

Default: 0

Models Supported: 1520, 4697, 4698, USB OEM, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

For scanners that support Interleaved 2-of-5 bar codes, this property can be used to indicate a second valid length for Interleaved 2-of-5 bar codes. When reading Interleaved 2-of-5 bar codes, these scanners can read only bar codes of one or two specific lengths. The setITFLength1 and setITFLength2 properties are used to specify the Interleaved 2-of-5 lengths to be recognized and read by these scanners. The value of the setITFLength2 property must be an even number from 4 to 32. This value specifies the exact length of the bar code. A value of 0 (zero) indicates that only Interleaved 2-of-5 bar codes of the length specified in the iTFLength1 resource are read by the scanner. If the scanner is not configured to read Interleaved 2-of-5 bar codes, the value of this property is ignored.

setITFLengths

Property Type: boolean

Default: false

Models Supported: USB OEM, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

Valid values:

true ITF lengths represent a range.

false ITF lengths are discrete.

Note: For the AnyPlace Kiosk Line scanner, *true* indicates that ITF lengths specifies a minimum value only (ITFLength1).

setITFLengthSpecifiedTwo

Property Type: boolean

Default: false

Models Supported: 4697, 4698, USB OEM

This property specifies if two ITF lengths are specified.

setITFScansPerRead

Property Type: byte

Default: 1

Models Supported:4698

This property controls the minimum number of scans performed for Interleaved 2-of-5 labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the default value should be used.

setLaserTimeOut

Property Type: byte

Default: 0

Models Supported: 4696, 4697, 4698, USB OEM

This property specifies the length of the period of inactivity that causes a laser scanner to turn off its laser. This property can have the following values for 4696, 4697 and 4698:

0 The laser turns off after 15 minutes of inactivity

5 The laser turns off after 5 minutes of inactivity

10 The laser turns off after 10 minutes of inactivity

15 The laser turns off after 15 minutes of inactivity

This property can have the following values for USB OEM:

X'00' The laser always stays on

X'08' The laser turns off after 5 minutes of inactivity

X'10' The laser turns off after 10 minutes of inactivity

X'18' The laser turns off after 15 minutes of inactivity

Note: For motor-driven laser scanners, the motor is always on if the laser is on.

setLED_GoodRead_Duration

Property Type: byte

Default: 0

Models Supported: USB OEM

This property specifies the LED good read duration This property can have the following values:

X'00' Use a short time value for LED good read duration

X'01' Use a medium time value for LED good read duration

X'02' Use a long time value for LED good read duration

X'03' Use the longest time value for LED good read duration

setMotorTimeOut

Property Type: byte

Default: 0

Models Supported: 4696, 4697, 4698, USB OEM

This property specifies the length of the period of inactivity that causes a motorized laser scanner to turn off its motor. This property can have the following values for 4696, 4697 and 4698:

0 The motor turns off after 60 minutes of inactivity

5 The motor turns off after 5 minutes of inactivity

10 The motor turns off after 10 minutes of inactivity

15 The motor turns off after 15 minutes of inactivity

30 The motor turns off after 30 minutes of inactivity

60 The motor turns off after 60 minutes of inactivity

This property can have the following values for USB OEM:

X'00' The motor is always on

X'01' The motor turns off after 5 minutes of inactivity

X'02' The motor turns off after 10 minutes of inactivity

X'03' The motor turns off after 15 minutes of inactivity

X'04' The motor turns off after 30 minutes of inactivity

X'05' The motor turns off after 60 minutes of inactivity

setSecurityLevelForInStore

Property Type: byte

Default: 0

Models Supported: USB OEM

This property specify the security/integrity level for In-Store labels. This property can have the following values:

X'00' Use a low security/integrity level.

X'01' Use a medium security/integrity level.

X'02' Use a high security/integrity level.

X'03' Use the highest security/integrity level.

setSTFLength1

Property Type: byte

Default: 0

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

This property allows you to set the first Standard 2 of 5 length to be accepted if STFLengths = false (discrete) or the minimum length to be accepted if STFLengths = true (range)

setSTFLength2

Property Type: byte

Default: 0

Models Supported: Line scanner, Omni scanner

This property allows you to set the second Standard 2 of 5 length to be accepted if STFLengths = false (discrete) If STFLengths = true (range) STFLength2 is the maximum value to be accepted for the Omni scanner. For the Line scanner, this value is meaningless.

setSTFLengths

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

Valid values:

- true (range)
  - For Line scanners, STF Length1 represent a minimum length to be accepted. STF Length2 is ignored.
  - For Omni scanners, STF Length1/2 represent a minimum/maximum length to be accepted.
- false (discrete) STF Length1 and Length2 are the only lengths to be accepted

Note: To accept any length STF, this property must be set to *true* and STFLength1 to 0.

setStoreScansPerRead

Property Type: byte

Default: 1

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for in-store labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the number of scans per read is not specified by the application.

setSupplementals

Property Type: boolean

Default: false

Models Supported: 4698, AnyPlace Kiosk Omni scanner

This property controls whether the scanner recognizes 2-digit supplementals and 5 digit supplementals for UPCA, UPCE, EAN/JAN8, AND EAN/JAN13. When enabled, supplementals are optional for all these symbologies.

setSupplementalsSecurityLevel

Property Type: byte

Default: 10

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

This property specify the security level for UPC-A, UPC-E, EAN/JAN-8, and EAN/JAN-13 supplementals. Making this value higher reduces the chance of ″short readings″ but increases the difficulty of reading low-quality barcodes. This property can have values in the range 0 - 100.

setTransmitScannerPrefixAndSuffix

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner.

Scanner data must be transmitted without any prefixes or suffixes. If they are required, set this property to true. Note that there is no property that allows to set them to a particular char sequence (such as suffix = CR), so they must be set using barcode programming

setUPCAScansPerRead

Property Type: byte

Default: 1

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for UPC-A labels. This is the number of scans for a single pass of an item over the scanner window. The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the number of scans per read is not specified by the application.

setUPCDScansPerRead

Property Type: byte

Default: 1

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for UPC-D labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the number of scans per read is not specified by the application.

setUPCEScansPerRead

Property Type: byte

Default: 2

Models Supported: 4696, 4697, 4698

This property controls the minimum number of scans performed for UPC-E labels. This is the number of scans for a single pass of an item over the scanner window.

The value of this property has a range from 0 (zero) to 4. Any value outside of this range is not valid. A value of 0 (zero) indicates that the number of scans per read is not specified by the application. This property can have the following values:

1 One scan

2 Two scans

3 Three scans

4 Four scans

Additional JavaPOS information

IBM 4697 Point of Sale Scanner Configuration

The IBM 4697 Point of Sale Scanner Model 1 follows the following rules in the symbology configuration:

1. No more than 2 of the supported symbology options may be included in the JPOS.XML entry for the scanner.
2. Only one of these two symbology options is allowed to be an industrial symbology. Choosing more than one industrial symbology option is considered invalid.
3. One of the non-industrial symbology must be included.

Non-industrial symbologies are considered to be:

- UPC-A/E, EAN/JAN-8/13
- UPC-A/E, EAN/JAN-8/13, UPC-D1..D5

Industrial symbologies are considered to be:

- ITF (Interleaved 2 of 5)

- CODABAR
- CODE 39
- CODE 93
- CODE 128

Example 1 - To enable one symbology with UPC-A/E, EAN/JAN-8/13, UPC-D1..D5 add following properties to the JposEntry:

```
<prop name="setEnableUPCAE_EANJAN813" type="Boolean" value="true"/>
<prop name="setEnableUPCD1D5" type="Boolean" value="true"/>
```

Example 2 - To enable two symbologies with UPC-A/E, EAN/JAN-8-13 and ITF add following properties to the JposEntry:

```
<prop name="setEnableUPCAE_EANJAN813" type="Boolean" value="true"/>
<prop name="setEnableInterleaved20f5" type="Boolean" value="true"/>
<prop name="setITFLength1" type="Byte" value="12"/>
```

USB OEM Scanner and scale configuration

The IBM JavaPOS supports new USB scanner and scale devices that comply with IBM or USB OEM Point-of-Sale Device interface specifications. Support for these devices required modification to configuration files. These configuration files currently include entries for known USB OEM Scanner devices. for a new device, the configuration files must be modified to include the VendorID and ProductID of that device.

For the Windows environment, two configuration files must be modified: javaxusb.inf and posj.properties.

For Linux, only posj.properties must be changed to include the VendorID and ProductID values.

Update javaxusb.inf (Windows only)

The javaxusb.inf file must be updated to include an entry for the new USB OEM Scanner device.

1. Obtain VendorID and ProductID for your device using a USB utility, such as UsbView.
2. Edit the javaxusb.inf file, located in the default location C:\POS\IBMJPOS\Lib\.
3. In the OEM Scanner section, copy the last entry and paste it to the end of that section.
4. Modify the USB\VID_XXXX&PID_YYYY part of the line by substituting the VendorID for XXXX, and the ProductID for YYYY.

To configure the new device:

1. Go to a command prompt.
2. Change directory to: C:\POS\IBMJPOS\Lib
3. Enter: **jxusbset.exe -IBMSETUP C:\POS\IBMJPOS\Lib\javaxusb.inf**

An example entry:
New entry at the end of OEM Scanner section for VID = 0x1234, PID = 0x5678
″Add a description for the scanner″=
JAVAXUSB_COMPOSITE, USB\VID_1234&PID_5678

Update posj.properties configuration file

For both Windows and Linux environments, the posj.properties file must be updated to include VendorID and ProductID for the USB OEM scanner.

1. Obtain VendorID and ProductID for your device using a USB utility, such as UsbView.
2. Edit the posj.properties file, located at:
   - for Windows: C:\pos\ibmjpos\config
   - for IRES: /opt/ibm/javapos/etc
3. In the OEM Scanner section, copy the last entry and paste it to the end of that section.
4. Modify the VendorID and the ProductID to match your device.
5. Increment the numeric digit of the entry.

For example, if your new scanner has Vendor ID 0x1234 and Product ID 0x5678, and the last entry was com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevice.72, then add the following entry:

```
com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevice.73 = 0x1234,0x5678
```

Known scanner issues

This section describes known issues found with specific models.

PSC 384 Magellan - USB

SetEnableCodabar and SetEnableCode93 configuration parameters must not be enabled. The scanner does not work correctly if these are enabled.

IBM 4698

- RS-485 through USB protocol converter
- SetEnableCodabar, SetEnableCode93, and SetEnableUCC_EAN128 configuration parameters must not be enabled. The scanner does not work correctly if these are enabled.
- Configuration must be correct for the scanner type attached, even though it looks like a USB scanner.

OPOS configuration

Scanner programming

RS-485 and USB scanner configuration can be modified programmatically using the resource file, AIPSYS.RES, as mentioned earlier in “Modifying device behavior (USB, RS-485 and PS/2 devices)” on page 50. This allows the scanner configuration to be changed without having to physically reprogram every scanner by hand using the manufacturer’s programming barcodes.

The resource file will probably not support every possible configuration option that a particular scanner supports. The resource file support is based on older IBM scanner hardware specifications and most scanner manufacturers have added features beyond those specifications. You may still have to use the manufacturer’s programming barcodes to enable/disable some options for your scanners.

There are many configuration options or resources that you can set in the resource file for the scanner. They are all documented in Chapter 21 ″Resource Sets″ of *Point of Sale Subsystem: Programming Reference and User’s Guide* (SC30-3560). The resources that are typically set for a scanner are:

*barCodes1
*barCodes2
*barCodes3
*barCodes4
*itfLength1
*itfLength2
*itfLengthType

The barCodesX resources determine what types of barcodes the scanner will read. The itfLengthXXX resources apply specifically to Interleaved 2 of 5 (ITF) barcodes and determine what subset of ITF barcodes the scanner will read.

Scanners are identified by the driver as being one of two types, hand-held or flatbed (table top). Each of these types is further divided into different subtypes based upon which IBM scanner interface is being emulated. For each IBM scanner interface, only certain resources can be used and for each resource, only certain values can be specified.

For example, an RS485 scanner configured for port 17B will report itself to the terminal (and have different configuration options) than the same physical scanner configured for port 5B.

If the resource file does not contain any scanner configuration information, the driver still sends a default configuration command to the scanner. See Table 201 and Table 202 for the default configuration for each scanner subtype.

Table 201. Handheld scanners

Subtype	Description	Subset of valid resources	Default configuration
1	IBM Hand Held Bar Code Reader model 1 (HHBCR1)	*barCode1	UPC-A, UPC-E, EAN-8, EAN-13, JAN-8, JAN-13
2	IBM Hand Held Bar Code Reader model 2 (HHBCR2)	*barCode1	UPC-A, UPC-E, EAN-8, EAN-13, JAN-8, JAN-13
3	IBM Model 1520	*barCode1 *barCode2 *barCode3 *barCode4 *itfLength1	UPC-A, UPC-E, UPC-D, UPC-8, UPC-13
4	USB scanner	*barCode1 *barCode2 *barCode3 *barCode4 *itfLength1 *itfLength2 *itfLengthType	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13

Table 202. Flatbed scanners

Subtype	Description	Subset of valid resources	Default configuration
1	IBM 4687 model 1 & 2	Cannot be programmed via the resource file	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13
2	IBM 4687 model 1 & 2	*barCode1 *barCode2 *itfLength1 *itfLength2	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13

Table 202. Flatbed scanners (continued)

Subtype	Description	Subset of valid resources	Default configuration
3	IBM 4686 model 3 & 4	*barCode1 *barCode2 *itfLength1 *itfLength2	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13
4	IBM 4696	*barCode1	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13
5	IBM 4697	*barCode1 *barCode2 *itfLength1 *itfLength2	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13
6	IBM 4698	*barCode1 *barCode2 *itfLength1 *itfLength2	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13
7	USB scanner	*barCode1 *barCode2 *barCode3 *barCode4 *itfLength1 *itfLength2 *itfLengthType	UPC-A, UPC-E, UPC-D, EAN-8, EAN-13, JAN-8, JAN-13

Following are several scanner configuration examples:

1. One IBM 1520-compatible hand-held scanner is attached and it must be able to read Code 39 and UPC barcodes. The resource file should contain the following:

   ```
   *barCodes1: LGROUP_UPC_EAN
   *barCodes2: LGROUP_CODE_39
   ```

2. One IBM USB OEM POS Device Interface-compliant scanner is attached and it needs to be able to read Code 39, Code 128, UPC, EAN and ITF barcodes (4 to 32 digits (inclusive) in length). The resource file should contain the following:

   ```
   *barCodes1: LGROUP_UPC_EAN_D1_TO_D5
   *barCodes2: LGROUP_CODE_39
   *barCodes3: LGROUP_INT_2_OF_5
   *itfLength1: 4
   *itfLength2: 32
   *itfLengthType: 1
   ```

3. One IBM USB OEM POS Device Interface-compliant scanner is attached and it needs to be bale to read Code 39, Code 128, UPC, EAN and ITF barcodes (20 and 24 digits in length only). The resource file should contain the following:

   ```
   *barCodes1: LGROUP_UPC_EAN_D1_TO_D5
   *barCodes2: LGROUP_CODE_39
   *barCodes3: LGROUP_INT_2_OF_5
   *itfLength1: 20
   *itfLength2: 24
   *itfLengthType: 0
   ```

4. One IBM HHBCR2-compatible hand-held scanner is attached and it needs to be able to read ITF barcodes. The resource file should contain the following:

   ```
   *barCodes1: LGROUP_INT_2_OF_5
   ```

5. One IBM HHBCR2-compatible hand-held scanner is attached and it needs to be able to read UPC, EAN and ITF barcodes. The resource file should contain the following:

```
*barCodes1: LGROUP_UPC_EAN_ITF
```

6. One IBM HHBCR2-compatible hand-held scanner is attached and it needs to be able to read Code 128, UPC, EAN and ITF barcodes. The resource file should contain the following:

   ```
   *barCodes1: 0
   ```

   Note: For IBM HHBCR2-compatible scanners, 0 (zero) specifies that the scanner should be placed in **test** mode which allows it to read all barcode types that it supports.

7. One RS485 IBM HHBCR2-compatible hand-held scanner (OPOS name, **HH485**) and one IBM USB OEM POS Device Interface-compliant flatbed scanner (OPOS name, **FBUSB**) are attached; both scanners must be able to read Code 39, Code 128, UPC, and EAN barcodes. The resource file should contain the following:

   ```
   *HH485.barCodes1: 0

   *FBUSB.barCodes1: LGROUP_UPC_EAN
   *FBUSB.barCodes1: LGROUP_CODE_39_CODE_128
   ```

8. One IBM 1520-compatible hand-held scanner is attached and it needs to be able to read Code 39 and 20-digit ITF barcodes. The resource file should contain the following:

   ```
   *barCodes1: LGROUP_UPC_EAN
   *barCodes2: LGROUP_INT_2_OF_5
   *itfLength1: 20
   ```

Additional OPOS information

Scanner detection

The IBM OPOS driver will control all RS-485 scanners, and can optionally control USB scanners.

RS-485 scanners are detected automatically by the driver, and do not need any special programming in order to be detected. The only programming that might be required is to alter the model of legacy IBM scanner that the physical scanner is emulating, since altering the model of scanner will change what configuration options you can use to program the scanner through the driver.

USB scanners will be detected automatically if the scanner has been programmed as an IBM-compatible device. If it has not been programmed as such, you must use the scanner vendors own scanner driver.

Some scanner vendors have implemented their driver with the scanner programmed as an IBM-compatible device. In this situation both the vendors own driver and the IBM driver will send programming commands to the device, since both drivers recognize the scanner as a device that they should control.

To work around this problem refer to “Ignoring non-IBM USB devices” on page 50.

Chapter 18. Scale

Supported devices

Table 203. Scale supported devices

Device	Connectivity	Comments
1. IBM 4687 Compatible Scanner/scale	RS-485, USB	
2. IBM 4696 Compatible Scanner/scale	RS-485, USB	
3. IBM 4698 Compatible Scanner/scale	RS-485, USB	
4. OEM Scale	USB	Must conform to *IBM USB OEM Point-of-Sale Device Interface Specification* V1.0 or later.

Supported properties and methods

Table 204. Scale common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	False
CheckHealthText	All	
Claimed	All	
DataCount	All	
DataEventEnabled	All	
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	Not supported	
PowerNotify	All	
PowerState	All	

Table 204. Scale common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 205. Scale specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	
CapDisplay	All	
CapDisplayText	Not supported	
CapPriceCalculating	Not supported	
CapStatusUpdate	False	
CapTareWeight	Not supported	
CapZeroScale	4	All except 1
MaxDisplayTextChars	Not supported - 0	
MaximumWeight	All	
SalesPrice	Not supported	
ScaleLiveWeight	Not supported	
StatusNotify	Not supported	
TareWeight	Not supported	
UnitPrice	Not supported	
WeightUnit	All	

Table 206. Scale common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearInput	All	
clearOutput	All	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 207. Scale specific methods

Method	JavaPOS and OPOS Gateway	OPOS
displayText	Not supported	
readWeight	All	
zeroScale	4	All except 1

Table 208. Scale events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	
DirectIOEvent	Not supported	Not supported
ErrorEvent	All	
StatusUpdateEvent	All	

JavaPOS configuration

Configuration options depend on the hardware capabilities. Table 209 details the configuration options supported by the scale models.

Table 209. Scale configuration options

Configuration option	IBM 4687	IBM 4698	USB OEM
centerOfZero		✓	✓
displayRequired		✓	✓
enforceZeroReturn		✓	✓
fiveDigitWeight			✓
operationMode		✓	✓
vibrationSensitivity		✓	✓
weighMode		✓	✓

CenterOfZero

```
<prop name="com.ibm.jpos.sdi.config.Scale.centerOfZero" type="Boolean" value="false"/>
```

Indicate center-of-zero with an LED.

Valid values are TRUE and FALSE. Default is FALSE.

DisplayRequired

```
<prop name="com.ibm.jpos.sdi.config.Scale.displayRequired" type="Boolean" value="false"/>
```

This property specifies whether a remote scale display is required. It is your responsibility to ensure that a system operating without a remote display meets the applicable weights and measures regulations.

If the value of this resource is false, but there is a remote display attached to the scale device, the display might not be used.

If the value of this resource is true but there is no remote display attached to the scale device, the scale might be unusable.

Valid values are TRUE and FALSE. Default is FALSE.

EnforceZeroReturn

```
<prop name="com.ibm.jpos.sdi.config.Scale.enforceZeroReturn" type="Boolean" value="false"/>
```

This property specifies whether zero protection is required. With zero protection enabled, the scale does not answer weight requests if a negative weight value is indicated on the display prior to placing the item for weighing on the scale, or an item is left on the scale for four minutes.

If either of these conditions exists and the value of this property is true, no weight data can be transmitted to the host until the scale returns to a zero weight (either by resetting the scale or removing the item).

Valid values are TRUE and FALSE. Default is FALSE

FiveDigitWeight (USB OEM only)

```
<prop name="com.ibm.jpos.sdi.config.Scale.fiveDigitWeight" type="Boolean" value="true"/>
```

This property specifies whether to return 5 digits (units of thousandths of pounds) or 4 digits (units of hundredths of pounds) for a non-metric weight.

Valid values are TRUE and FALSE. Default is TRUE.

OperationMode

```
<prop name="com.ibm.jpos.sdi.config.Scale.operationMode" type="String" value="US"/>
```

This property specifies the regulations that the point-of-sale system must conform to. The weight and measures requirements in various countries differ, and require minor operational differences.

Valid values are US and UK. Default is US.

US The scale conforms to regulations specified by the United States NIST Handbook 44 and the Canadian Department of Consumer and Corporate Affairs, Weights and Measurements Act, Specifications SGM-1.

UK The scale conforms to regulations specified by the Non-automatic Weighing Instruments (EEC Requirements) Regulations 1992 (based on OIML R 76-1).

VibrationSensitivity

```
<prop name="com.ibm.jpos.sdi.config.Scale.vibrationSensitivity" type="String" value="0"/>
```

This property controls the scale's sensitivity to vibration. External vibrations can affect the stability of the scale. There is a programmable vibration filter that allows you to reduce the scale's sensitivity to vibration. Reducing the scale's sensitivity to

vibration increases the scale settling time slightly, so the higher vibration filter settings should be selected only when testing reveals a stability problem in the checkstand.

Valid values are 0 (normal sensitivity) to 3 (Ultra low sensitivity). Default is 0.

Default Value: 0

WeighMode

```
<prop name="com.ibm.jpos.sdi.config.Scale.weighMode" type="String" value="0"/>
```

This property determines whether the weight is returned in pounds (Avoirdupois or English system) or in kilograms (Metric system).

Valid values are 0 and 1. Default is 0.

0 Weight is given in pounds. For USB OEM compliant scales the fiveDigitWeight property specifies the number of digits to return for an English weight (pounds). For 4698/4696 scales, four digits represent the weight of the item in hundredths.

1 Weight is given in kilograms. All scales return five digits which represent the weight of the item in thousandths of kilograms.

Additional JavaPOS information

Known scale issues

This section describes known issues with specific models.

PSC 384 Magellan - USB
This model does not support the fiveDigitWeight configuration property. This property must be set to *false*; otherwise the Scale configuration fails. For more information, see “FiveDigitWeight (USB OEM only)” on page 248.

OPOS configuration

Table 210. Service Object settings for scale

Keyword	Type	Description	OPOS Gateway Support
weightMode	String	Unit of weight used by the scale. Valid values are: **0** U.S. pound (default) **1** Kilogram	No
RemoteDisplayAttached	String	Whether a Pole Display is attached. Valid values are: **True** Display attached **(other value)** No display (default)	No
ZeroScale	String	Whether zeroing the scale is supported. Valid values are: **0** Not supported (default) **1** Supported	No

Table 210. Service Object settings for scale (continued)

Keyword	Type	Description	OPOS Gateway Support
NCRCompatible	String	Determines how zero weight is handled. Valid values are: **False** Working according to UnifiedPOS specification **True** When stable weight of zero, return OPOS_SUCCESS	No

Chapter 19. Tone indicator

Tone indicator supported devices

Table 211. Tone indicator supported devices

Device	Connectivity
1. Retail alphanumeric POS keyboard w/Card Reader	PS/2, RS-485, USB
2. Retail alphanumeric POS Keyboard with MSR and pointing device	PS/2
3. Retail POS keyboard	RS-485, USB
4. Retail POS keyboard with Card Reader	RS-485, USB
5. Retail POS keyboard w/Card Reader and Display	RS-485, USB
6. Modifiable Layout Keyboard w/Card Reader	RS-485, USB
7. SurePoint 4820 integrated Tone Indicator	RS-485, USB, PS/2
8. SureOne Integrated Keyboard	PS/2
9. Compact Alphanumeric POS Keyboard	PS/2
10. POS keyboard V	RS-485, USB
11. PC POS Keyboard or PC Point of Sale Keyboard	PS/2, RS-485, USB
12. PLU keyboard/Display III	RS-485, USB
13. Retail POS keyboard VI	RS-485, USB
14. SurePOS 100 Integrated keyboard	PS/2
15. 4674 Integrated Keyboard	RS-485
16. 4685-KC1 (OPOS only)	RS-485
17. 4685-K01 (OPOS only)	RS-485
18. 4685–K02 (Ultra7) keyboard with card reader	RS-485, USB
19. 4685-K02 with MSR/Encoder and 4 position Keylock	USB
20. 4685-K02 with MSR/Encoder and 6 position Keylock	USB
21. 4685-K03	RS-485, USB
22. 4610 SST Tx6/Tx7	EIA-232, RS-485, USB
23. Internal speaker (OPOS only)	Integrated
22. Retail alphanumeric POS keyboard Version 2	PS/2, USB
23. Compact Alphanumeric POS Keyboard Version 2	PS/2, USB
24. Retail POS Modular 67 Key Keyboard	PS/2, USB

Supported properties and methods

Table 212. Tone indicator common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	
CapPowerReporting	PR_STANDARD	All support PR_STANDARD

Table 212. Tone indicator common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapStatisticsReporting	True (see Appendix A, "JavaPOS support for UnifiedPOS device statistics properties," on page 373)	False
CapUpdateFirmware	False	
CapUpdateStatistics	False (see Appendix A, "JavaPOS support for UnifiedPOS device statistics properties," on page 373)	False
CheckHealthText	All	
Claimed	All	
ClearInput	Not supported	
DataCount	Not supported	
DataEventEnabled	All	Not supported
DeviceControlDescription	All	
DeviceControlVersion	All	
DeviceEnabled	All	
DeviceServiceDescription	All	
DeviceServiceVersion	All	
FreezeEvents	All	
OpenResult	Not supported	All
OutputID	All	
PowerNotify	All	
PowerState	All	
PhysicalDeviceDescription	All	
PhysicalDeviceName	All	
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	

Table 213. Tone indicator specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	
CapPitch	All	
CapVolume	All	
InterToneWait	All	
Tone1Duration	All	
Tone1Pitch	All	
Tone1Volume	All	
Tone2Duration	All	
Tone2Pitch	All	
Tone2Volume	All	

Table 214. Tone indicator common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	
claim	All	
clearOutput	All	
close	All	
compareFirmwareVersion	Not supported	
directIO	Not supported	
open	All	
release	All	
resetStatistics	Not supported	
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 373)	Not supported
updateFirmware	Not supported	
updateStatistics	Not supported	

Table 215. Tone indicator specific methods

Method	JavaPOS and OPOS Gateway	OPOS
sound	All	
soundImmediate	All	

Table 216. Tone indicator events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	Not supported	
ErrorEvent	All	
OutputCompleteEvent	All	
StatusUpdateEvent	All	

OPOS configuration

Table 217. Service Object settings for tone indicator

Keyword	Type	Description	OPOS Gateway Support
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.	No

Part 3. Keyboards and code pages

Chapter 20. POS keyboard layouts and scan codes

This chapter contains keyboard layouts and scan codes for IBM Point-of-Sale keyboards.

Note: The numeric keypad is shaded in all illustrations.

Understanding scan codes

When a key is pressed on any keyboard (sometimes referred to as, a KEY_DOWN event), the keyboard device driver receives a code that is called a *make scan code*. There is a different code for each key. When a key is released (on some keyboards) sometimes referred to as a KEY_UP event, the keyboard device driver receives a code that is called a *break scan code*. These codes are translated into ASCII character codes by the operating system.

The following scancode tables do not contain values for the OPOS driver, because the OPOS driver only sends a data event to the application if each individual key has been mapped in the configuration program. If a key has been mapped by the user the value that will be sent to the application will be the user-defined mapped value, not the hardware scancode.

For keys that generate different scan codes based on a modifier key (IE, S1 and S2) the scan code shown in the table below does not include the make scan code or break scan code for the modifier key, only the scan code for that key.

Checkout Keyboards Layout (50-key and Modular 67 Key)

This section contains illustrations of the layouts for the following keyboards:

- Retail Point of Sale Keyboard (50-key)
- Retail Point of Sale Keyboard with card reader (50-key)
- Retail Point of Sale Keyboard with card reader and display (50-key)
- Point of Sale Keyboard VI
- Modular 67 Key Keyboard

50-Key Modifiable Layout Keyboard

Figure 33 shows the key-switch numbers. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

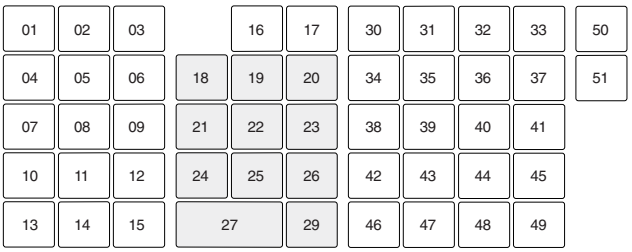


Figure 33. Layout for 50-Key Modifiable Keyboard

50-Key Modifiable Layout Keyboard and Operator Display

Figure 34 shows the key-switch numbers. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

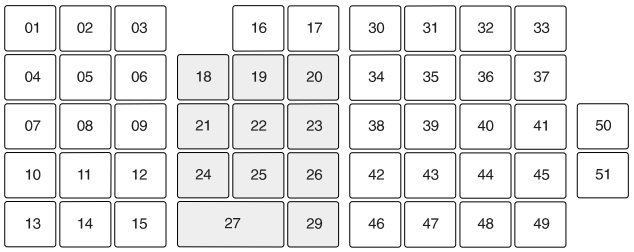


Figure 34. Layout for 50-Key Modifiable Keyboard and Operator Display

Retail Point of Sale Keyboard Layout

Figure 35 on page 261 shows the key-switch numbers for the Retail Point of Sale Keyboard, both with and without the card reader. The numeric keypad (key switches 18 to 29) are shown in the shaded area of the illustration.

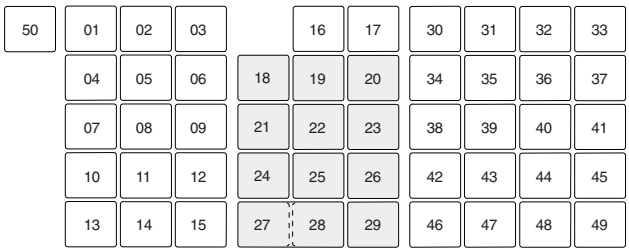


Figure 35. Layout for Retail Point of Sale Keyboard

Retail Point of Sale Keyboard with Card Reader and Display

Figure 36 shows the key-switch numbers for the Retail Point of Sale Keyboard with Card Reader and Display. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

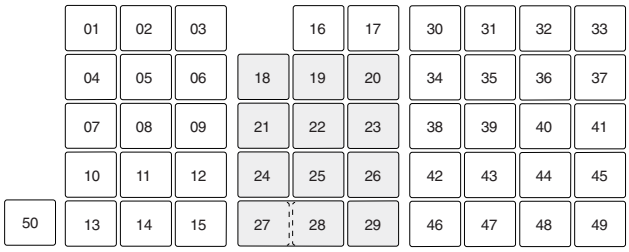


Figure 36. Layout for Retail Point of Sale Keyboard with Card Reader and Display

Point of Sale Keyboard VI Layout

Figure 37 shows the key-switch numbers for the Point of Sale Keyboard VI. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

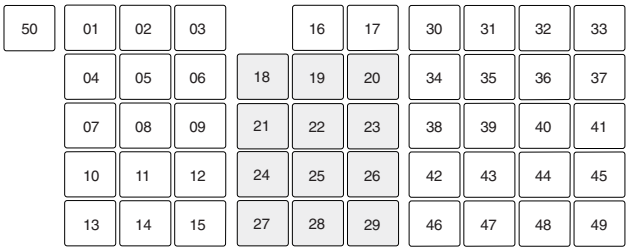


Figure 37. Layout for Point of Sale Keyboard VI

Modular 67 Key Keyboard Layout

Figure 38 on page 262 shows the key-switch numbers for the Modular 67 Key Keyboard. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

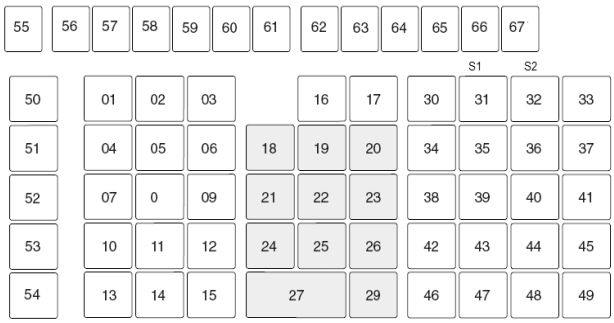


Figure 38. Layout for Modular 67 Key Keyboard

Checkout Keyboards PS/2 scan code set

Table 218 shows the key scan codes for the Modular 67 Key Keyboard.

Table 218. PS/2 scan codes for Modular 67 Key Keyboard

Key switch number	Hardware make code	Hardware break code	JavaPOS – POSKeyData	Comments
1	X'3b'	X'bb'	X'3b'	
2	X'3c'	X'bc'	X'3c'	
3	X'3d'	X'bd'	X'3d'	
4	X'3e'	X'be'	X'3e'	
5	X'3f'	X'bf'	X'3f'	
6	X'40'	X'c0'	X'40'	
7	X'41'	X'c1'	X'41'	
8	X'42'	X'c2'	X'42'	
9	X'43'	X'c3'	X'43'	
10	X'44'	X'c4'	X'44'	
11	X'57'	X'd7'	X'57'	
12	X'58'	X'd8'	X'58'	
13	X'0f'	X'8f'	X'0f'	
14	X'e0' X'5b'	X'e0' X'db'	X'01' X'5b'	
15	X'38'	X'b8'	X'38'	
16	X'46'	X'c6'	X'46'	
17	X'45'	X'c5'	X'45'	
18	X'47'	X'c7'	X'47'	
19	X'48'	X'c8'	X'48'	
20	X'49'	X'c9'	X'49'	
21	X'4b'	X'cb'	X'4b'	
22	X'4c'	X'cc'	X'4c'	
23	X'4d'	X'cd'	X'4d'	
24	X'4f'	X'cf'	X'4f'	

Table 218. PS/2 scan codes for Modular 67 Key Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS – POSKeyData	Comments
25	X'50'	X'd0'	X'50'	
26	X'51'	X'd1'	X'51'	
27	X'64'	X'e4'	X'64'	
28	X'52'	X'd2'	X'52'	
29	X'53'	X'd2'	X'53'	
30	X'29'	X'a9'	X'29'	
31	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
32	X'e1' X'1d' X'45' X'e1' X'9d' X'c5'		X'01' X'45'	Key press only
33	X'0e'	X'8e'	X'0e'	
34	X'0c'	X'8c'	X'0c'	
35	X'0d'	X'8d'	X'0d'	
36	X'28'	X'a8'	X'28'	
37	X'2b'	X'ab'	X'2b'	
38	X'33'	X'b3'	X'33'	
39	X'34'	X'b4'	X'34'	
40	X'35'	X'b35'	X'35'	
41	X'27'	X'a7'	X'27'	
42	X'65'	X'e5'	X'65'	
43	X'66'	X'e6'	X'66'	
44	X'67'	X'e7'	X'67'	
45	X'68'	X'e8'	X'68'	
46	X'1d'	X'9d'	X'1d'	
47	X'39'	X'b9'	X'39'	
48	X'2a'	X'aa'	X'2a'	
49	X'e0' X'1c'	X'e0' X'9c'	X'01' X'1c'	
50	X'01'	X'81'	X'01'	
51	X'1e'	X'9e'	X'1e'	
52	X'30'	X'b0'	X'30'	
53	X'2e'	X'ae'	X'2e'	
54	X'20'	X'a0'	X'20'	
55	X'12'	X'92'	X'12'	
56	X'21'	X'a1'	X'21'	
57	X'22'	X'a2'	X'22'	
58	X'23'	X'a3'	X'23'	
59	X'17'	X'97'	X'17'	
60	X'24'	X'a4'	X'24'	
61	X'25'	X'a5'	X'25'	
62	X'26'	X'a6'	X'26'	

Table 218. PS/2 scan codes for Modular 67 Key Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS – POSKeyData	Comments
63	X'32'	X'b2'	X'32'	
64	X'31'	X'b1'	X'31'	
65	X'18'	X'98'	X'18'	
66	X'19'	X'99'	X'19'	
67	X'10'	X'90'	X'10'	
	+Ctrl Case pressed			
31	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
32	X'e1' X'1d' X'45' X'e1' X'9d' X'c5'		X'01' X'45'	Key press only

Checkout keyboards RS-485/USB scan code set

Table 219 shows the key scan codes

- 50-Key Modifiable Layout Keyboard
- 50-Key Modifiable Layout Keyboard and Operator Display
- Retail Point of Sale Keyboard (50-Key)
- Retail Point of Sale Keyboard with card reader (50-key)
- Retail Point of Sale Keyboard with card reader and display (50-key)
- Point of Sale Keyboard VI
- Modular 67 Key Keyboard

The hardware scan code set for the 50-Key Modifiable Layout Keyboard, and the 50-Key Modifiable Layout Keyboard and Operator Display is different from the hardware scan code set for the retail point-of-sale keyboards. In order to allow the application to work with either keyboard more easily, the hardware scan code set for the 50-Key Modifiable Layout Keyboard and the 50-Key Modifiable Layout Keyboard and Operator Display is translated to the scan code set for the retail point-of-sale keyboards.

Note: The 50-Key Modifiable Layout Keyboard and 50-Key Modifiable Layout Keyboard and Operator Display only generate make scan codes when a key is pressed. In order for your application to work with either the old or the new keyboards, it must use the make scan codes from the checkout style keyboards and discard any break scan codes it receives.

Table 219. Checkout keyboards RS-485/USB scan code set

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'4b'	X'f0' X'4b'	X'4b'	
2	X'3b'	X'f0' X'3b'	X'3b'	
3	X'6b'	X'f0' X'6b'	X'6b'	
4	X'4c'	X'f0' X'4c'	X'4c'	
5	X'3c'	X'f0' X'3c'	X'3c'	
6	X'6c'	X'f0' X'6c'	X'6c'	

Table 219. Checkout keyboards RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
7	X'4f'	X'f0' X'4f'	X'4f'	
8	X'3f'	X'f0' X'3f'	X'3f'	
9	X'6f'	X'f0' X'6f'	X'6f'	
10	X'4e'	X'f0' X'4e'	X'4e'	
11	X'3e'	X'f0' X'3e'	X'3e'	
12	X'6e'	X'f0' X'6e'	X'6e'	
13	X'4d'	X'f0' X'4d'	X'4d'	
14	X'3d'	X'f0' X'3d'	X'3d'	
15	X'6d'	X'f0' X'6d'	X'6d'	
16	X'7b'	X'f0' X'7b'	X'7b'	
17	X'1b'	X'f0' X'1b'	X'1b'	
18	X'7c'	X'f0' X'7c'	X'7c'	
19	X'0c'	X'f0' X'0c'	X'0c'	
20	X'1c'	X'f0' X'1c'	X'1c'	
21	X'7f'	X'f0' X'7f'	X'7f'	
22	X'0f'	X'f0' X'0f'	X'0f'	
23	X'1f'	X'f0' X'1f'	X'1f'	
24	X'7e'	X'f0' X'7e'	X'7e'	
25	X'0e'	X'f0' X'0e'	X'0e'	
26	X'1e'	X'f0' X'1e'	X'1e'	
27	X'7d'	X'f0' X'7d'	X'7d'	
28	X'0d'	X'f0' X'0d'	X'0d'	
29	X'1d'	X'f0' X'1d'	X'1d'	
30	X'8b'	X'f0' X'8b'	X'8b'	
31	X'af'	X'f0' X'af'	X'af'	
32	X'bf'	X'f0' X'bf'	X'bf'	
33	X'bb'	X'f0' X'bb'	X'bb'	
34	X'8c'	X'f0' X'8c'	X'8c'	
35	X'9c'	X'f0' X'9c'	X'9c'	
36	X'ac'	X'f0' X'ac'	X'ac'	
37	X'bc'	X'f0' X'bc'	X'bc'	
38	X'8f'	X'f0' X'8f'	X'8f'	
39	X'9f'	X'f0' X'9f'	X'9f'	
40	X'9b'	X'f0' X'9b'	X'9b'	
41	X'ab'	X'f0' X'ab'	X'ab'	
42	X'8e'	X'f0' X'8e'	X'8e'	
43	X'9e'	X'f0' X'9e'	X'9e'	
44	X'ae'	X'f0' X'ae'	X'ae'	
45	X'be'	X'f0' X'be'	X'be'	

Table 219. Checkout keyboards RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
46	X'8d'	X'f0' X'8d'	X'8d'	
47	X'9d'	X'f0' X'9d'	X'9d'	
48	X'ad'	X'f0' X'ad'	X'ad'	
49	X'bd'	X'f0' X'bd'	X'bd'	
50	X'50'	X'f0' X'50'	X'50'	
51	X'5A'	X'F0'X'5A'	X'5A'	67–Key
52	X'2B'	X'F0'X'2B'	X'2B'	67–Key
53	X'61'	X'F0'X'61'	X'61'	67–Key
54	X'62'	X'F0'X'62'	X'62'	67–Key
55	X'21'	X'F0'X'21'	X'21'	67–Key
56	X'52'	X'F0'X'52'	X'52'	67–Key
57	X'53'	X'F0'X'53'	X'53'	67–Key
58	X'32'	X'F0'X'32'	X'32'	67–Key
59	X'23'	X'F0'X'23'	X'23'	67–Key
60	X'2A'	X'F0'X'2A'	X'2A'	67–Key
61	X'28'	X'F0'X'28'	X'28'	67–Key
62	X'08'	X'F0'X'08'	X'08'	67–Key
63	X'56'	X'F0'X'56'	X'56'	67–Key
64	X'07'	X'F0'X'07'	X'07'	67–Key
65	X'1A'	X'F0'X'1A'	X'1A'	67–Key
66	X'63'	X'F0'X'63'	X'63'	67–Key
67	X'64'	X'F0'X'64'	X'64'	67–Key
	+ Ctrl Case Pressed			
31	X'f0' X'50' X'00' X'50'		X'00'	Key press only
32	X'f0' X'50' X'01' X'50'		X'01'	Key press only

Note: 67–Key

These keys are present only on the Modular 67 Key Keyboard.

Table 220 on page 267 explains the scan codes received for the S1 and S2 function keys on the retail point-of-sale keyboards.

The S1 and S2 function keys send a series of scan codes on the retail point-of-sale keyboards. These function keys generate a break scan code for the Ctrl key (scan code of 0x50 with PosKC_KEYUP flag set), a make scan code for the S1 or S2 key and then a make scan code for the Ctrl key (scan code of 0x50 with PosKC_KEYUP flag reset). The S1 and S2 function keys can only be accessed by pressing the Ctrl key. With the Ctrl key pressed, key 31 represents the S1 function and key 32 represents the S2 function.

No break scan codes are sent for the 0x00 or 0x01 scan codes.

Table 220. RS-485/USB scan codes for the Retail Point of Sale Keyboards

Key switch number	Key type	Ctrl + scan code
31	S1	X'00'
32	S2	X'01'

Modifiable-layout keyboard with card reader layout (133-key)

Figure 39 shows the key-switch numbers for the modifiable layout keyboard with card reader. The three possible locations for the numeric keypad are shown in the shaded area of the illustration. The default location for the numeric keypad is the right-most shaded area.

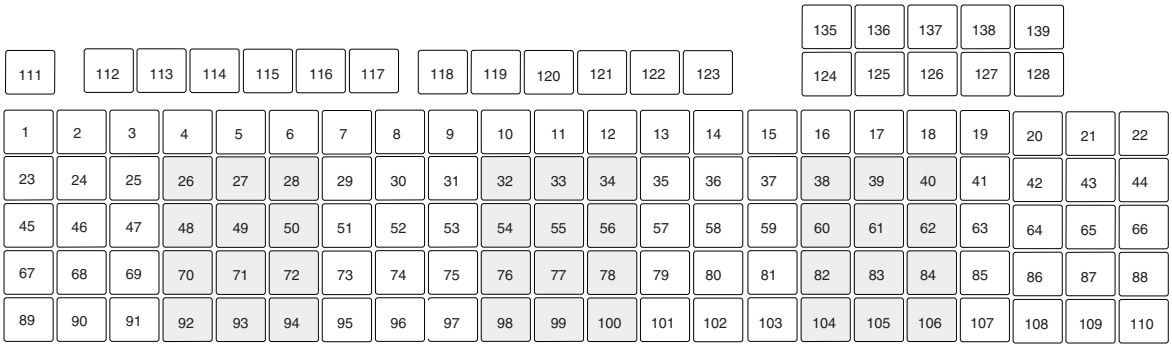


Figure 39. Modifiable layout keyboard with card reader

Modifiable layout keyboard with card reader RS-485/USB scan code set

Table 221 shows the key scan codes for the modifiable layout keyboard with card reader.

Table 221. Modifiable layout keyboard RS-485/USB scan code set

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'b4'	X'f0' X'b4'	X'b4'	
2	X'a4'	X'f0' X'a4'	X'a4'	
3	X'34'	X'f0' X'34'	X'34'	
4	X'44'	X'f0' X'44'	X'44'	
5	X'94'	X'f0' X'94'	X'94'	
6	X'84'	X'f0' X'84'	X'84'	
7	X'14'	X'f0' X'14'	X'14'	
8	X'04'	X'f0' X'04'	X'04'	
9	X'74'	X'f0' X'74'	X'74'	
10	X'64'	X'f0' X'64'	X'64'	
11	X'54'	X'f0' X'54'	X'54'	
12	X'5b'	X'f0' X'5b'	X'5b'	
13	X'6b'	X'f0' X'6b'	X'6b'	
14	X'7b'	X'f0' X'7b'	X'7b'	
15	X'0b'	X'f0' X'0b'	X'0b'	
16	X'1b'	X'f0' X'1b'	X'1b'	
17	X'8b'	X'f0' X'8b'	X'8b'	
18	X'9b'	X'f0' X'9b'	X'9b'	
19	X'3b'	X'f0' X'3b'	X'3b'	

Table 221. Modifiable layout keyboard RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
20	X'4b'	X'f0' X'4b'	X'4b'	
21	X'ab'	X'f0' X'ab'	X'ab'	
22	X'bb'	X'f0' X'bb'	X'bb'	
23	X'b5'	X'f0' X'b5'	X'b5'	
24	X'a5'	X'f0' X'a5'	X'a5'	
25	X'35'	X'f0' X'35'	X'35'	
26	X'45'	X'f0' X'45'	X'45'	
27	X'95'	X'f0' X'95'	X'95'	
28	X'85'	X'f0' X'85'	X'85'	
29	X'15'	X'f0' X'15'	X'15'	
30	X'05'	X'f0' X'05'	X'05'	
31	X'75'	X'f0' X'75'	X'75'	
32	X'65'	X'f0' X'65'	X'65'	
33	X'55'	X'f0' X'55'	X'55'	
34	X'5a'	X'f0' X'5a'	X'5a'	
35	X'6a'	X'f0' X'6a'	X'6a'	
36	X'7a'	X'f0' X'7a'	X'7a'	
37	X'0a'	X'f0' X'0a'	X'0a'	
38	X'1a'	X'f0' X'1a'	X'1a'	
39	X'8a'	X'f0' X'8a'	X'8a'	
40	X'9a'	X'f0' X'9a'	X'9a'	
41	X'3a'	X'f0' X'3a'	X'3a'	
42	X'4a'	X'f0' X'4a'	X'4a'	
43	X'aa'	X'f0' X'aa'	X'aa'	
44	X'ba'	X'f0' X'ba'	X'ba'	
45	X'b6'	X'f0' X'b6'	X'b6'	
46	X'a6'	X'f0' X'a6'	X'a6'	
47	X'36'	X'f0' X'36'	X'36'	
48	X'46'	X'f0' X'46'	X'46'	
49	X'96'	X'f0' X'96'	X'96'	
50	X'86'	X'f0' X'86'	X'86'	
51	X'16'	X'f0' X'16'	X'16'	
52	X'06'	X'f0' X'06'	X'06'	
53	X'76'	X'f0' X'76'	X'76'	
54	X'66'	X'f0' X'66'	X'66'	
55	X'56'	X'f0' X'56'	X'56'	
56	X'59'	X'f0' X'59'	X'59'	
57	X'69'	X'f0' X'69'	X'69'	
58	X'79'	X'f0' X'79'	X'79'	

Table 221. Modifiable layout keyboard RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
59	X'09'	X'f0' X'09'	X'09'	
60	X'19'	X'f0' X'19'	X'19'	
61	X'89'	X'f0' X'89'	X'89'	
62	X'99'	X'f0' X'99'	X'99'	
63	X'39'	X'f0' X'39'	X'39'	
64	X'49'	X'f0' X'49'	X'49'	
65	X'a9'	X'f0' X'a9'	X'a9'	
66	X'b9'	X'f0' X'b9'	X'b9'	
67	X'b3'	X'f0' X'b3'	X'b3'	
68	X'a3'	X'f0' X'a3'	X'a3'	
69	X'33'	X'f0' X'33'	X'33'	
70	X'43'	X'f0' X'43'	X'43'	
71	X'93'	X'f0' X'93'	X'93'	
72	X'83'	X'f0' X'83'	X'83'	
73	X'13'	X'f0' X'13'	X'13'	
74	X'03'	X'f0' X'03'	X'03'	
75	X'73'	X'f0' X'73'	X'73'	
76	X'63'	X'f0' X'63'	X'63'	
77	X'53'	X'f0' X'53'	X'53'	
78	X'5c'	X'f0' X'5c'	X'5c'	
79	X'6c'	X'f0' X'6c'	X'6c'	
80	X'7c'	X'f0' X'7c'	X'7c'	
81	X'0c'	X'f0' X'0c'	X'0c'	
82	X'1c'	X'f0' X'1c'	X'1c'	
83	X'8c'	X'f0' X'8c'	X'8c'	
84	X'9c'	X'f0' X'9c'	X'9c'	
85	X'3c'	X'f0' X'3c'	X'3c'	
86	X'4c'	X'f0' X'4c'	X'4c'	
87	X'ac'	X'f0' X'ac'	X'ac'	
88	X'bc'	X'f0' X'bc'	X'bc'	
89	X'b2'	X'f0' X'b2'	X'b2'	
90	X'a2'	X'f0' X'a2'	X'a2'	
91	X'32'	X'f0' X'32'	X'32'	
92	X'42'	X'f0' X'42'	X'42'	
93	X'92'	X'f0' X'92'	X'92'	
94	X'82'	X'f0' X'82'	X'82'	
95	X'12'	X'f0' X'12'	X'12'	
96	X'02'	X'f0' X'02'	X'02'	
97	X'72'	X'f0' X'72'	X'72'	

Table 221. Modifiable layout keyboard RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
98	X'62'	X'f0' X'62'	X'62'	
99	X'52'	X'f0' X'52'	X'52'	
100	X'5d'	X'f0' X'5d'	X'5d'	
101	X'6d'	X'f0' X'6d'	X'6d'	
102	X'7d'	X'f0' X'7d'	X'7d'	
103	X'0d'	X'f0' X'0d'	X'0d'	
104	X'1d'	X'f0' X'1d'	X'1d'	
105	X'8d'	X'f0' X'8d'	X'8d'	
106	X'9d'	X'f0' X'9d'	X'9d'	
107	X'3d'	X'f0' X'3d'	X'3d'	
108	X'4d'	X'f0' X'4d'	X'4d'	
109	X'ad'	X'f0' X'ad'	X'ad'	
110	X'bd'	X'f0' X'bd'	X'bd'	
111	X'20'	X'f0' X'20'	X'20'	
112	X'b1'	X'f0' X'b1'	X'b1'	
113	X'a1'	X'f0' X'a1'	X'a1'	
114	X'31'	X'f0' X'31'	X'31'	
115	X'41'	X'f0' X'41'	X'41'	
116	X'91'	X'f0' X'91'	X'91'	
117	X'81'	X'f0' X'81'	X'81'	
118	X'88'	X'f0' X'88'	X'88'	
119	X'18'	X'f0' X'18'	X'18'	
120	X'58'	X'f0' X'58'	X'58'	
121	X'68'	X'f0' X'68'	X'68'	
122	X'78'	X'f0' X'78'	X'78'	
123	X'08'	X'f0' X'08'	X'08'	
124	X'1e'	X'f0' X'1e'	X'1e'	
125	X'8e'	X'f0' X'8e'	X'8e'	
126	X'9e'	X'f0' X'9e'	X'9e'	
127	X'ae'	X'f0' X'ae'	X'ae'	
128	X'be'	X'f0' X'be'	X'be'	
135	X'1f'	X'f0' X'1f'	X'1f'	
136	X'8f'	X'f0' X'8f'	X'8f'	
137	X'9f'	X'f0' X'9f'	X'9f'	
138	X'af'	X'f0' X'af'	X'af'	
139	X'bf'	X'f0' X'bf'	X'bf'	
	+ Ctrl Case Pressed			
124 (S2)	X'f0' X'20' X'01' X'20'		X'01'	Key press only

Table 221. Modifiable layout keyboard RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
135 (S1)	X'f0' X'20' X'00' X'20'		X'00'	Key press only

Table 222 explains the scan codes received for the S1 and S2 function keys on the Modifiable Layout Keyboard with Card Reader.

The S1 and S2 function keys send a series of scan codes on the Modifiable Layout Keyboard with Card Reader. These function keys generate a break scan code for the Ctrl key (scan code of 0x20 with PosKC_KEYUP flag set), a make scan code for the S1 or S2 key, and then a make scan code for the Ctrl key (scan code of 0x20 with the PosKC_KEYUP flag reset). The S1 and S2 function keys can only be accessed by pressing the **Ctrl** key. With the **Ctrl** key pressed, key 135 represents the S1 function, and key 124 represents the S2 function.

No break scan codes are sent for the 0x00 or 0x01 scan codes.

Table 222. RS-485 Scan Codes – Modifiable Layout Keyboard with Card Reader

Key switch number	Key type	Ctrl + scan code
124	S2	X'01'
135	S1	X'00'

Retail Alphanumeric Point of Sale keyboard with card reader (RS-485 or USB attached) and Modular Alphanumeric Point of Sale Keyboard and Modular Compact Alphanumeric Point of Sale Keyboard (USB attached)

This section contains illustrations of the layouts for the following keyboards:

- PC Point of Sale Keyboard with Card Reader (ANKPOS)
- Retail Alphanumeric POSKeyboard with Card Reader (NANPOS)
- Modular Alphanumeric POS Keyboard
- Modular Compact Alphanumeric POS Keyboard

PC Point of Sale keyboard with card reader (ANKPOS)

Figure 40 shows the layout and assigned key-switch numbers for the PC Point of Sale Keyboard. The numeric keypad location is shown in the shaded area of the illustration.

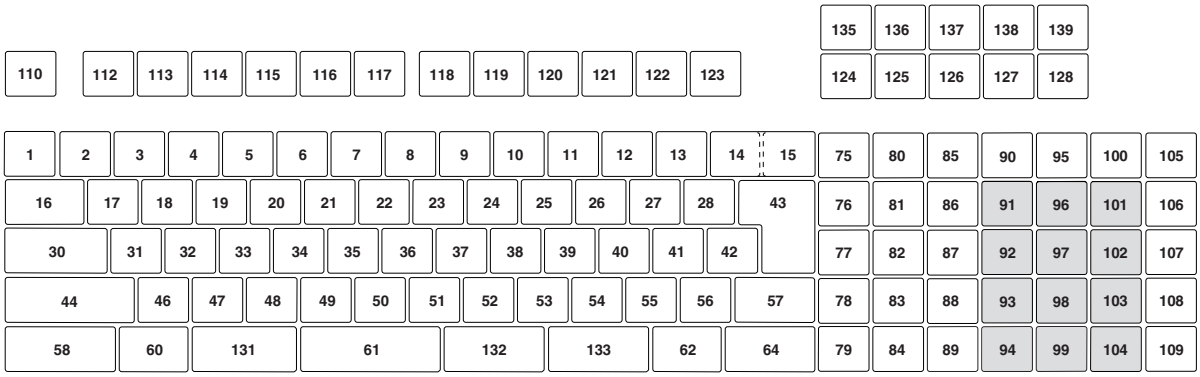


Figure 40. PC Point of Sale keyboard layout

Retail Alphanumeric Point of Sale Keyboard with card reader (NANPOS)

Figure 41 shows the layout and assigned key-switch numbers for the Retail Alphanumeric Point of Sale Keyboard with card reader.

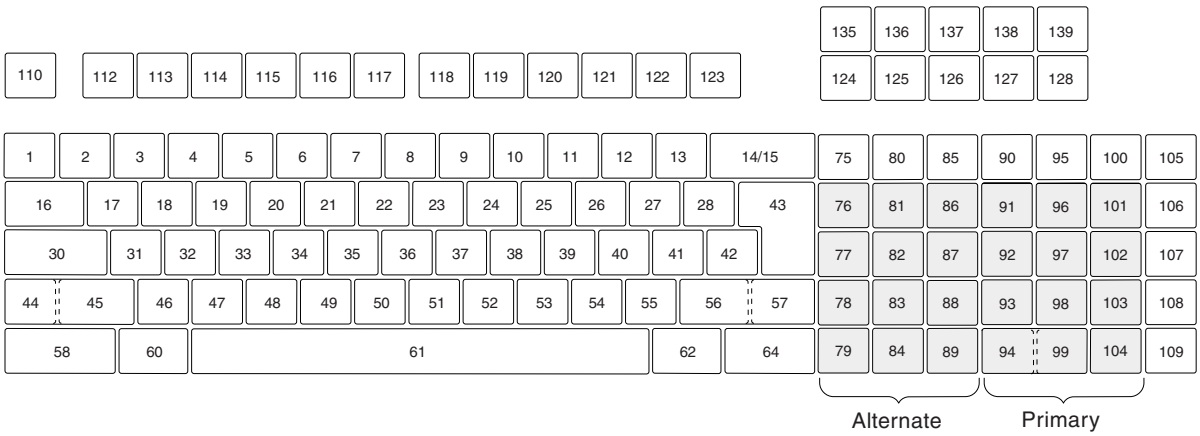


Figure 41. Layout and Assigned Switch Numbers

Notes:

1. The two possible locations for the numeric keypad are shown in the shaded area of the illustration. The default location for the numeric keypad is the rightmost shaded area.
2. Key 45 appears as a single key only on non-U.S. keyboards. On U.S. keyboards, key 44 also covers key 45 (key 44, 45 is a double key).
3. Keys 94 and 99 can have a single, horizontal double-wide key covering both keys, or they can be split into two individual keys.
4. This keyboard is similar to the 101-enhanced keyboard and the 102-enhanced keyboard. The following keys are on the Retail Alphanumeric Point of Sale Keyboard with Card Reader, but not on the 101-enhanced keyboard or the 102-enhanced keyboard. These keys are referred to as the *point-of-sale-unique keys*.

77	106
78	107
82	108
87	124
88	125
90	126
95	127
99	128
100	135
105	

See Table 223 on page 275 for the scan codes associated with these keys.

Modular Alphanumeric POS Keyboard

Figure 42 shows the layout and assigned key-switch numbers for the Modular Alphanumeric POS Keyboard.

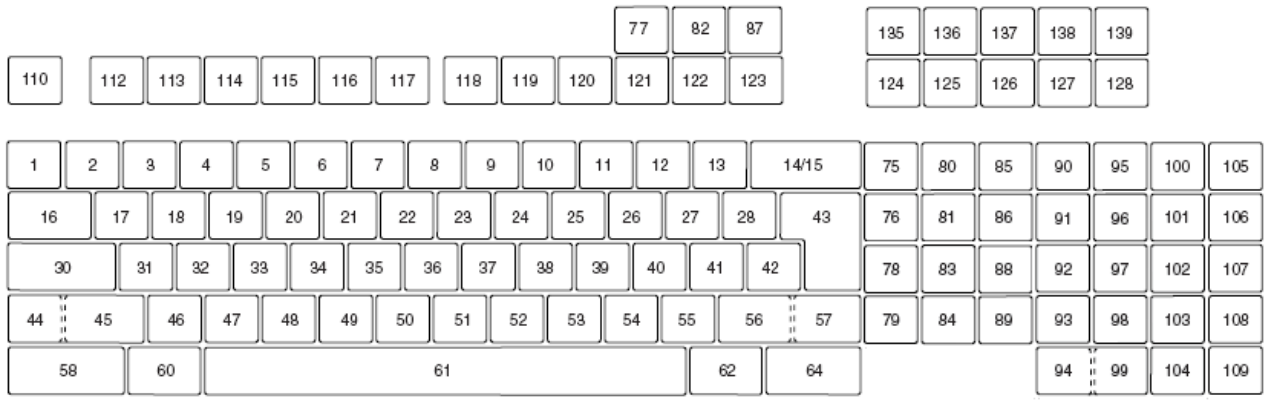


Figure 42. Layout and Assigned Switch Numbers for Modular Alphanumeric POS Keyboard

See Table 223 on page 275 for the scan codes associated with these keys.

Modular Compact Alphanumeric Point of Sale Keyboard

Figure 43 shows the layout and assigned key-switch numbers for the Modular Compact Alphanumeric POS Keyboard.

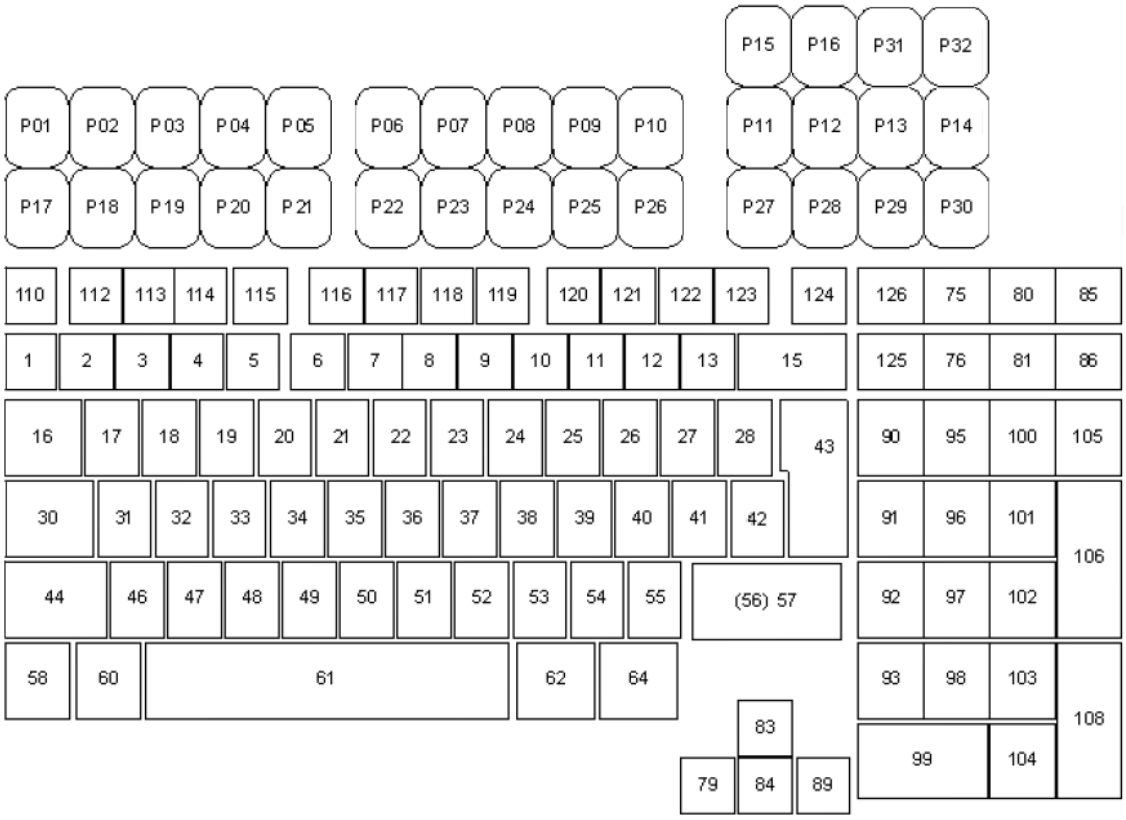


Figure 43. Layout and Assigned Switch Numbers for the Modular Compact Alphanumeric POS Keyboard

See Table 226 on page 280 for the scan codes associated with these keys.

RS-485/USB scan code set for the Retail Alphanumeric Point of Sale Keyboard (NANPOS), PC Point of Sale Keyboard (ANKPOS) and USB scan code set for Modular Alphanumeric Point of Sale Keyboard

Table 223 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the RS-485 or USB port.

Table 223. RS-485/USB scan code set

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'51'	X'f0' X'51'	X'51'	
2	X'11'	X'f0' X'11'	X'11'	
3	X'12'	X'f0' X'12'	X'12'	
4	X'13'	X'f0' X'13'	X'13'	
5	X'14'	X'f0' X'14'	X'14'	

Table 223. RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
6	X'54'	X'f0' X'54'	X'54'	
7	X'55'	X'f0' X'55'	X'55'	
8	X'15'	X'f0' X'15'	X'15'	
9	X'18'	X'f0' X'18'	X'18'	
10	X'16'	X'f0' X'16'	X'16'	
11	X'17'	X'f0' X'17'	X'17'	
12	X'57'	X'f0' X'57'	X'57'	
13	X'58'	X'f0' X'58'	X'58'	
14	X'6A'	X'f0' X'6A'	X'6A'	ANKPOS (see Notes)
15	X'7A'	X'f0' X'7A'	X'7A'	
16	X'71'	X'f0' X'71'	X'71'	
17	X'61'	X'f0' X'61'	X'61'	
18	X'62'	X'f0' X'62'	X'62'	
19	X'63'	X'f0' X'63'	X'63'	
20	X'64'	X'f0' X'64'	X'64'	
21	X'74'	X'f0' X'74'	X'74'	
22	X'75'	X'f0' X'75'	X'75'	
23	X'65'	X'f0' X'65'	X'65'	
24	X'68'	X'f0' X'68'	X'68'	
25	X'66'	X'f0' X'66'	X'66'	
26	X'67'	X'f0' X'67'	X'67'	
27	X'77'	X'f0' X'77'	X'77'	
28	X'78'	X'f0' X'78'	X'78'	
30	X'72'	X'f0' X'72'	X'72'	
31	X'81'	X'f0' X'81'	X'81'	
32	X'82'	X'f0' X'82'	X'82'	
33	X'83'	X'f0' X'83'	X'83'	
34	X'84'	X'f0' X'84'	X'84'	
35	X'24'	X'f0' X'24'	X'24'	
36	X'25'	X'f0' X'25'	X'25'	
37	X'85'	X'f0' X'85'	X'85'	
38	X'88'	X'f0' X'88'	X'88'	
39	X'86'	X'f0' X'86'	X'86'	
40	X'87'	X'f0' X'87'	X'87'	
41	X'27'	X'f0' X'27'	X'27'	
42	X'47'	X'f0' X'47'	X'47'	
43	X'4a'	X'f0' X'4a'	X'4a'	
44	X'79'	X'f0' X'79'	X'79'	

Table 223. RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
45	X'22'	X'f0' X'22'	X'22'	World Trade key (see Notes)
46	X'41'	X'f0' X'41'	X'41'	
47	X'42'	X'f0' X'42'	X'42'	
48	X'43'	X'f0' X'43'	X'43'	
49	X'44'	X'f0' X'44'	X'44'	
50	X'34'	X'f0' X'34'	X'34'	
51	X'35'	X'f0' X'35'	X'35'	
52	X'45'	X'f0' X'45'	X'45'	
53	X'48'	X'f0' X'48'	X'48'	
54	X'46'	X'f0' X'46'	X'46'	
55	X'37'	X'f0' X'37'	X'37'	
56	X'38'	X'f0' X'38'	X'38'	ANKPOS (see Notes)
57	X'49'	X'f0' X'49'	X'49'	
58	X'50'	X'f0' X'50'	X'50'	
60	X'2d'	X'f0' X'2d'	X'2d'	
61	X'3a'	X'f0' X'3a'	X'3a'	
62	X'3d'	X'f0' X'3d'	X'3d'	
64	X'40'	X'f0' X'40'	X'40'	
75	X'5c'	X'f0' X'5c'	X'5c'	
76	X'5b'	X'f0' X'5b'	X'5b'	
77	X'5a'	X'f0' X'5a'	X'5a'	
78	X'2b'	X'f0' X'2b'	X'2b'	
79	X'3e'	X'f0' X'3e'	X'3e'	
80	X'4b'	X'f0' X'4b'	X'4b'	
81	X'1e'	X'f0' X'1e'	X'1e'	
82	X'7b'	X'f0' X'7b'	X'7b'	
83	X'2e'	X'f0' X'2e'	X'2e'	
84	X'3b'	X'f0' X'3b'	X'3b'	
85	X'8f'	X'f0' X'8f'	X'8f'	
86	X'1f'	X'f0' X'1f'	X'1f'	
87	X'7e'	X'f0' X'7e'	X'7e'	
88	X'7f'	X'f0' X'7f'	X'7f'	
89	X'3c'	X'f0' X'3c'	X'3c'	
90	X'9b'	X'f0' X'9b'	X'9b'	
91	X'6b'	X'f0' X'6b'	X'6b'	
92	X'0b'	X'f0' X'0b'	X'0b'	
93	X'8b'	X'f0' X'8b'	X'8b'	
94	X'bb'	X'f0' X'bb'	X'bb'	

Table 223. RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
95	X'9c'	X'f0' X'9c'	X'9c'	
96	X'6c'	X'f0' X'6c'	X'6c'	
97	X'0c'	X'f0' X'0c'	X'0c'	
98	X'8c'	X'f0' X'8c'	X'8c'	
99	X'bc'	X'f0' X'bc'	X'bc'	
100	X'ae'	X'f0' X'ae'	X'ae'	
101	X'6f'	X'f0' X'6f'	X'6f'	
102	X'0f'	X'f0' X'0f'	X'0f'	
103	X'5f'	X'f0' X'5f'	X'5f'	
104	X'4f'	X'f0' X'4f'	X'4f'	
105	X'3f'	X'f0' X'3f'	X'3f'	
106	X'6e'	X'f0' X'6e'	X'6e'	
107	X'0e'	X'f0' X'0e'	X'0e'	
108	X'8e'	X'f0' X'8e'	X'8e'	
109	X'be'	X'f0' X'be'	X'be'	
110	X'21'	X'f0' X'21'	X'21'	
112	X'52'	X'f0' X'52'	X'52'	
113	X'53'	X'f0' X'53'	X'53'	
114	X'32'	X'f0' X'32'	X'32'	
115	X'23'	X'f0' X'23'	X'23'	
116	X'2a'	X'f0' X'2a'	X'2a'	
117	X'28'	X'f0' X'28'	X'28'	
118	X'08'	X'f0' X'08'	X'08'	
119	X'56'	X'f0' X'56'	X'56'	
120	X'07'	X'f0' X'07'	X'07'	
121	X'1a'	X'f0' X'1a'	X'1a'	
122	X'1b'	X'f0' X'1b'	X'1b'	
123	X'1c'	X'f0' X'1c'	X'1c'	
124	X'1d'	X'f0' X'1d'	X'1d'	
125	X'6d'	X'f0' X'6d'	X'6d'	
126	X'4e'	X'f0' X'4e'	X'4e'	
127	X'4c'	X'f0' X'4c'	X'4c'	
128	X'9e'	X'f0' X'9e'	X'9e'	
131	X'31'	X'f0' X'31'	X'31'	ANKPOS (see Notes)
132	X'26'	X'f0' X'26'	X'26'	ANKPOS (see Notes)
133	X'36'	X'f0' X'36'	X'36'	ANKPOS (see Notes)
135	X'8d'	X'f0' X'8d'	X'8d'	
136	X'ac'	X'f0' X'ac'	X'ac'	
137	X'ab'	X'f0' X'ab'	X'ab'	

Table 223. RS-485/USB scan code set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
138	X'af'	X'f0' X'af'	X'af'	
139	X'bf'	X'f0' X'bf'	X'bf'	
	+ Left Ctrl Case Pressed			
124 (S2)	X'f0' X'50' X'01' X'50'		X'01'	Key press only (ANKPOS keylock in system position)
135 (S1)	X'f0' X'50' X'00' X'50'		X'00'	Key press only (ANKPOS keylock in system position)
	+Right Ctrl Case Pressed			
124 (S2)	X'f0' X'40' X'01' X'40'		X'01'	Key press only (ANKPOS keylock in system position)
135 (S1)	X'f0' X'40' X'00' X'40'		X'00'	Key press only (ANKPOS keylock in system position)
	+ Both Ctrl Case Pressed			
124 (S2)	X'f0' X'40' X'f0' X'50' X'01' X'50'		X'01'	Key press only (ANKPOS keylock in system position)
135 (S1)	X'f0' X'40' X'f0' X'50' X'00' X'50'		X'00'	Key press only (ANKPOS keylock in system position)
	+ Left Ctrl Case Pressed			
124 (S2)	X'f0' X'50' X'1d' X'50'		X'1d'	ANKPOS keylock in operator/manager position
135 (S1)	X'f0' X'50' X'8d' X'50'		X'8d'	ANKPOS keylock in operator/manager position
	+Right Ctrl Case Pressed			
124 (S2)	X'f0' X'40' X'1d' X'40'		X'1d'	ANKPOS keylock in operator/manager position
135 (S1)	X'f0' X'40' X'8d' X'40'		X'8d'	ANKPOS keylock in operator/manager position
	+ Both Ctrl Case Pressed			
124 (S2)	X'f0' X'40' X'f0' X'50' X'1d' X'50'		X'1d'	ANKPOS keylock in operator/manager position
135 (S1)	X'f0' X'40' X'f0' X'50' X'8d' X'50'		X'8d'	ANKPOS keylock in operator/manager position

Notes:

ANKPOS
PC Point of Sale Keyboard unique keys. These keys are present only on the PC Point of Sale (ANKPOS keyboard).

World Trade key
This key is present on all non-U.S. versions of the NANPOS keyboard, but not on the ANKPOS keyboard.

Table 224 explains the scan codes received for the S1 and S2 function keys on the Retail Alphanumeric Point of Sale Keyboard with Card Reader.

The S1 and S2 function keys send a series of scan codes on the Retail Alphanumeric Point of Sale Keyboard with Card Reader. These function keys generate a break scan code for the Ctrl key pressed (scan code 0x50 or 0x40 with PosKC_KEYUP flag set), a make scan code for the S1 or S2 key and then a make scan code for the Ctrl key pressed (scan code of 0x50 or 0x40 with PosKC_KEYUP flag reset). The S1 and S2 function keys can only be accessed by pressing the Ctrl key. With the Ctrl key pressed, key 135 represents the S1 function, and key 124 represents the S2 function.

No break scan codes are sent for the 0x00 or 0x01 scan codes.

Table 224. Point of Sale Scan Codes for Retail Alphanumeric Point of Sale Keyboard with Card Reader

Key switch number	Key type	Ctrl + scan code
124	S2	X'01'
135	S1	X'00'

For the PC Point of Sale Keyboard, Table 225 shows the scan codes sent by the S1 and S2 function keys, depending upon the Keylock Positions.

Table 225. Serial I/O Scan Codes– PC Point of Sale Keyboard (ANKPOS) Keyboard

Key switch number	Key type	Ctrl + scan code (keylock in system position)	Ctrl + scan code (keylock in operator/manager position)
124	S2	X'01'	X'1D'
135	S1	X'00'	X'8D'

USB scan code set for the Modular Compact Alphanumeric Point of Sale Keyboard

Table 226. USB scan codes for Modular Compact Alphanumeric POS Keyboard

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'51'	X'f0' X'51'	X'51'	
2	X'11'	X'f0' X'11'	X'11'	
3	X'12'	X'f0' X'12'	X'12'	
4	X'13'	X'f0' X'13'	X'13'	
5	X'14'	X'f0' X'14'	X'14'	
6	X'54'	X'f0' X'54'	X'54'	
7	X'55'	X'f0' X'55'	X'55'	
8	X'15'	X'f0' X'15'	X'15'	
9	X'18'	X'f0' X'18'	X'18'	
10	X'16'	X'f0' X'16'	X'16'	
11	X'17'	X'f0' X'17'	X'17'	

Table 226. USB scan codes for Modular Compact Alphanumeric POS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
12	X'57'	X'f0' X'57'	X'57'	
13	X'58'	X'f0' X'58'	X'58'	
15	X'7a'	X'f0' X'7a'	X'7a'	
16	X'71'	X'f0' X'71'	X'71'	
17	X'61'	X'f0' X'61'	X'61'	
18	X'62'	X'f0' X'62'	X'62'	
19	X'63'	X'f0' X'63'	X'63'	
20	X'64'	X'f0' X'64'	X'64'	
21	X'74'	X'f0' X'74'	X'74'	
22	X'75'	X'f0' X'75'	X'75'	
23	X'65'	X'f0' X'65'	X'65'	
24	X'68'	X'f0' X'68'	X'68'	
25	X'66'	X'f0' X'66'	X'66'	
26	X'67'	X'f0' X'67'	X'67'	
27	X'77'	X'f0' X'77'	X'77'	
28	X'78'	X'f0' X'78'	X'78'	
30	X'72'	X'f0' X'72'	X'72'	
31	X'81'	X'f0' X'81'	X'81'	
32	X'82'	X'f0' X'82'	X'82'	
33	X'83'	X'f0' X'83'	X'83'	
34	X'84'	X'f0' X'84'	X'84'	
35	X'24'	X'f0' X'24'	X'24'	
36	X'25'	X'f0' X'25'	X'25'	
37	X'85'	X'f0' X'85'	X'85'	
38	X'88'	X'f0' X'88'	X'88'	
39	X'86'	X'f0' X'86'	X'86'	
40	X'87'	X'f0' X'87'	X'87'	
41	X'27'	X'f0' X'27'	X'27'	
42	X'47'	X'f0' X'47'	X'47'	
43	X'4a'	X'f0' X'4a'	X'4a'	
44	X'79'	X'f0' X'79'	X'79'	
46	X'41'	X'f0' X'41'	X'41'	
47	X'42'	X'f0' X'42'	X'42'	
48	X'43'	X'f0' X'43'	X'43'	
49	X'44'	X'f0' X'44'	X'44'	
50	X'34'	X'f0' X'34'	X'34'	
51	X'35'	X'f0' X'35'	X'35'	
52	X'45'	X'f0' X'45'	X'45'	
53	X'48'	X'f0' X'48'	X'48'	

Table 226. USB scan codes for Modular Compact Alphanumeric POS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
54	X'46'	X'f0' X'46'	X'46'	
55	X'37'	X'f0' X'37'	X'37'	
57	X'49'	X'f0' X'49'	X'49'	
58	X'50'	X'f0' X'50'	X'50'	
60	X'2d'	X'f0' X'2d'	X'2d'	
61	X'3a'	X'f0' X'3a'	X'3a'	
62	X'3d'	X'f0' X'3d'	X'3d'	
64	X'40'	X'f0' X'40'	X'40'	
75	X'5c'	X'f0' X'5c'	X'5c'	
76	X'5b'	X'f0' X'5b'	X'5b'	
79	X'3e'	X'f0' X'3e'	X'3e'	
80	X'4b'	X'f0' X'4b'	X'4b'	
81	X'1e'	X'f0' X'1e'	X'1e'	
83	X'2e'	X'f0' X'2e'	X'2e'	
84	X'3b'	X'f0' X'3b'	X'3b'	
85	X'8f'	X'f0' X'8f'	X'8f'	
86	X'1f'	X'f0' X'1f'	X'1f'	
89	X'3c'	X'f0' X'3c'	X'3c'	
90	X'bf'	X'f0' X'bf'	X'bf'	
91	X'6b'	X'f0' X'6b'	X'6b'	
92	X'0b'	X'f0' X'0b'	X'0b'	
93	X'8b'	X'f0' X'8b'	X'8b'	
95	X'4f'	X'f0' X'4f'	X'4f'	
96	X'6c'	X'f0' X'6c'	X'6c'	
97	X'0c'	X'f0' X'0c'	X'0c'	
98	X'8c'	X'f0' X'8c'	X'8c'	
99	X'bb'	X'f0' X'bb'	X'bb'	
100	X'ae'	X'f0' X'ae'	X'ae'	
101	X'6f'	X'f0' X'6f'	X'6f'	
102	X'0f'	X'f0' X'0f'	X'0f'	
103	X'5f'	X'f0' X'5f'	X'5f'	
104	X'9c'	X'f0' X'9c'	X'9c'	
105	X'3f'	X'f0' X'3f'	X'3f'	
106	X'6e'	X'f0' X'6e'	X'6e'	
108	X'be'	X'f0' X'be'	X'be'	
110	X'21'	X'f0' X'21'	X'21'	
112	X'52'	X'f0' X'52'	X'52'	
113	X'53'	X'f0' X'53'	X'53'	
114	X'32'	X'f0' X'32'	X'32'	

Table 226. USB scan codes for Modular Compact Alphanumeric POS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
115	X'23'	X'f0' X'23'	X'23'	
116	X'2a'	X'f0' X'2a'	X'2a'	
117	X'28'	X'f0' X'28'	X'28'	
118	X'08'	X'f0' X'08'	X'08'	
119	X'56'	X'f0' X'56'	X'56'	
120	X'07'	X'f0' X'07'	X'07'	
121	X'1a'	X'f0' X'1a'	X'1a'	
122	X'1b'	X'f0' X'1b'	X'1b'	
123	X'1c'	X'f0' X'1c'	X'1c'	
124	X'ac'	X'f0' X'ac'	X'ac'	
125	X'ab'	X'f0' X'ab'	X'ab'	
126	X'af'	X'f0' X'af'	X'af'	
P01	X'4c'	X'f0' X'4c'	X'4c'	
P02	X'9e'	X'f0' X'9e'	X'9e'	
P03	X'4b'	X'f0' X'4b'	X'4b'	
P04	X'3b'	X'f0' X'3b'	X'3b'	
P05	X'6b'	X'f0' X'6f'	X'6b'	
P06	X'4c'	X'f0' X'4c'	X'4c'	
P07	X'3c'	X'f0' X'3c'	X'3c'	
P08	X'6c'	X'f0' X'6c'	X'6c'	
P09	X'4f'	X'f0' X'4f'	X'4f'	
P10	X'3f'	X'f0' X'3f'	X'3f'	
P11	X'1d'	X'f0' X'1d'	X'1d'	
P12	X'6f'	X'f0' X'6f'	X'6f'	
P13	X'4e'	X'f0' X'4e'	X'4e'	
P14	X'3e'	X'f0' X'3e'	X'3e'	
P15	X'8d'	X'f0' X'8d'	X'8d'	
P16	X'6e'	X'f0' X'6e'	X'6e'	
P17	X'5a'	X'f0' X'5a'	X'5a'	
P18	X'2b'	X'f0' X'2b'	X'2b'	
P19	X'7b'	X'f0' X'7b'	X'7b'	
P20	X'7e'	X'f0' X'7e'	X'7e'	
P21	X'7f'	X'f0' X'7f'	X'7f'	
P22	X'7c'	X'f0' X'7c'	X'7c'	
P23	X'7d'	X'f0' X'7d'	X'7d'	
P24	X'0d'	X'f0' X'0d'	X'0d'	
P25	X'9b'	X'f0' X'9b'	X'9b'	
P26	X'bc'	X'f0' X'bc'	X'bc'	
P27	X'0e'	X'f0' X'0e'	X'0e'	

Table 226. USB scan codes for Modular Compact Alphanumeric POS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
P28	X'8e'	X'f0' X'8e'	X'8e'	
P29	X'6d'	X'f0' X'6d'	X'6d'	
P30	X'4e'	X'f0' X'4e'	X'4e'	
P31	X'4d'	X'f0' X'4d'	X'4d'	
P32	X'3d'	X'f0' X'3d'	X'3d'	

Retail Alphanumeric Point of Sale keyboard with card reader (PS/2 or USB system attached)

This section contains illustrations of the layouts for the following keyboards:

- Retail Alphanumeric POSKeyboard with card reader (ANKPOS)
- Retail Alphanumeric POSKeyboard with card reader (NANPOS)
- Compact Alphanumeric POSKeyboard (CANPOS)
- Modular Alphanumeric POS Keyboard
- Modular Compact Alphanumeric POS Keyboard

Retail Alphanumeric Point of Sale with card reader or PC Point of Sale keyboard (ANKPOS) layout keyboard

Figure 44 shows the layout and assigned key-switch numbers for the PC Point of Sale keyboard. The numeric keypad location is shown in the shaded area of the illustration.

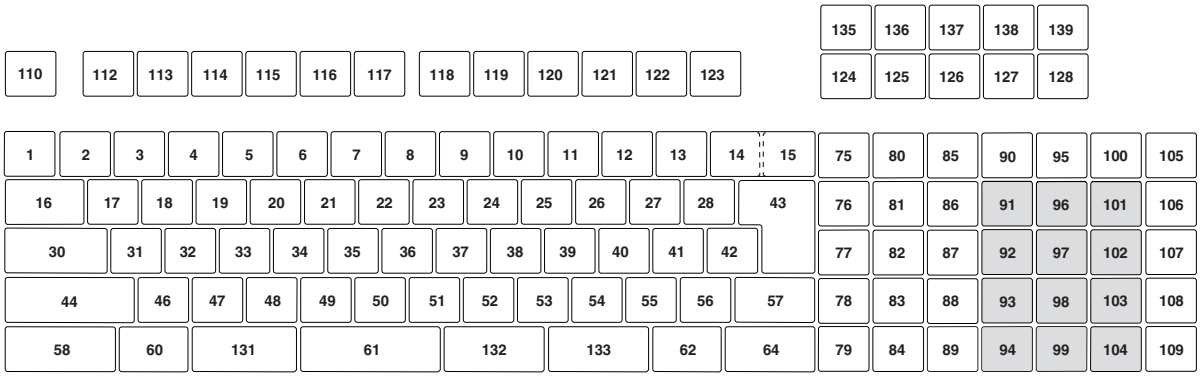


Figure 44. PC Point of Sale keyboard layout

Retail Alphanumeric Point of Sale with card reader (NANPOS) layout keyboard

Figure 45 shows the layout and assigned key-switch numbers for the Retail Alphanumeric Point of Sale Keyboard with Card Reader.

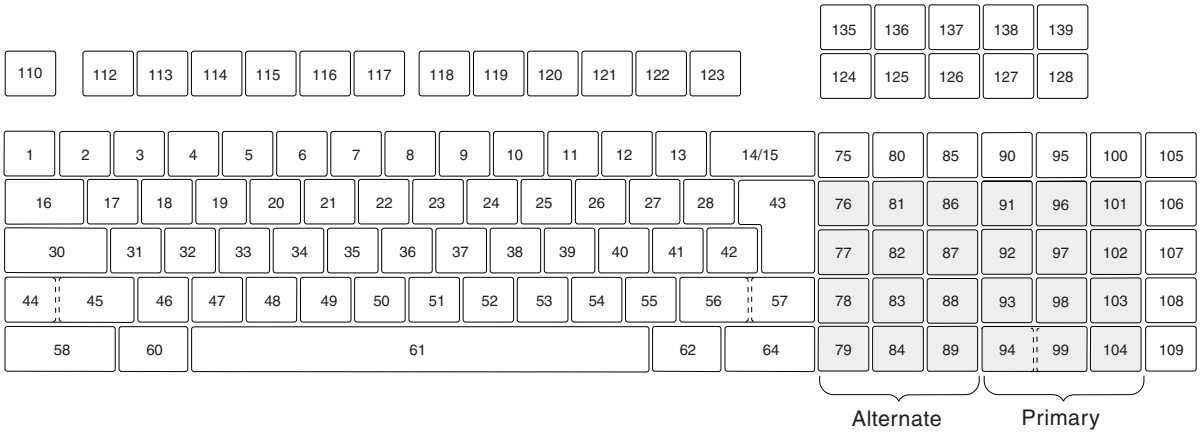


Figure 45. Layout and assigned switch numbers

Notes:

1. The two possible locations for the numeric keypad are shown in the shaded area of the illustration. The default location for the numeric keypad is the right-most shaded area.
2. Key 45 appears as a single key only on non-U.S. keyboards. On U.S. keyboards, key 44 also covers key 45 (key 44, 45 is a double key).
3. Keys 94 and 99 can have a single, horizontal double-wide key covering both keys, or they can be split into two individual keys.
4. This keyboard is similar to the 101-enhanced keyboard and the 102-enhanced keyboard. The following keys are on the Retail Alphanumeric Point of Sale Keyboard with Card Reader, but not on the 101-enhanced keyboard or the 102-enhanced keyboard. These keys are referred to as the *point-of-sale-unique keys*.

77	106
78	107
82	108
87	124
88	125
90	126
95	127
99	128
100	135
105	

See Table 227 on page 289 for the scan codes associated with these keys.

Modular Alphanumeric Point of Sale Keyboard

Figure 46 shows the layout and assigned key-switch numbers for the Modular Alphanumeric Point of Sale Keyboard.

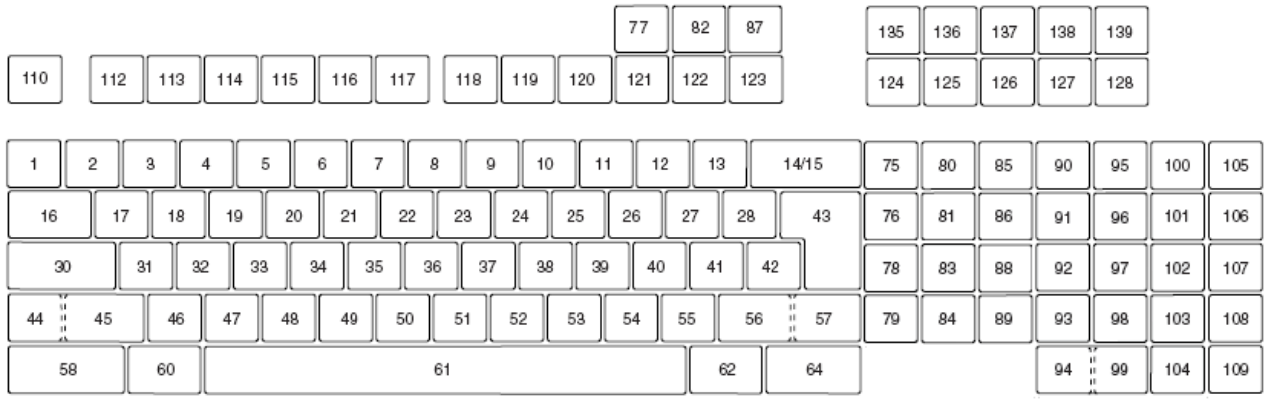


Figure 46. Layout and Assigned Switch Numbers for Modular ANPOS Keyboard

See Table 227 on page 289 for the scan codes associated with these keys.

Compact Alphanumeric Point of Sale (CANPOS) layout keyboard

Figure 47 shows the layout and assigned key-switch numbers for the Compact Alphanumeric Point of Sale keyboard with card reader.

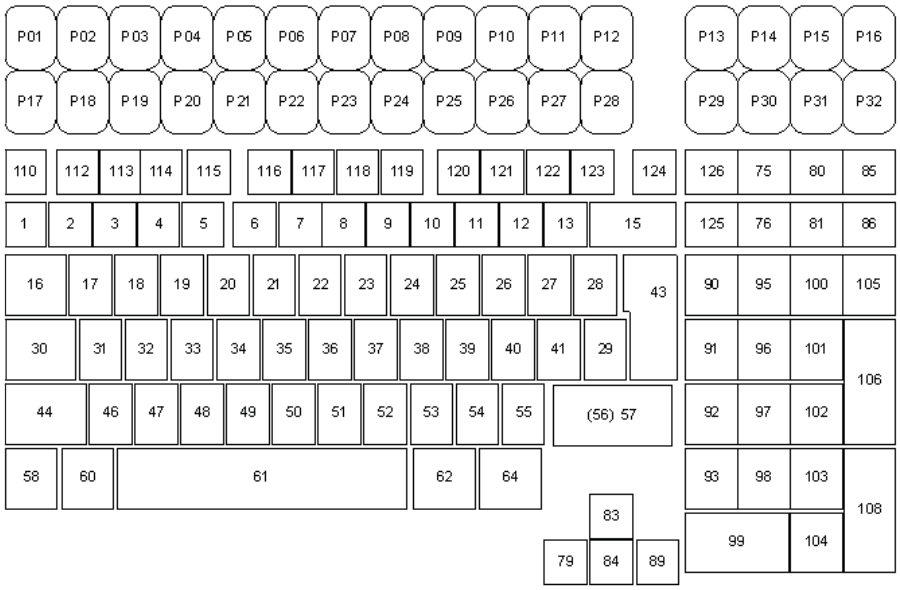


Figure 47. Layout and assigned switch numbers for CANPOS keyboard

Note: See Table 228 on page 296 for scan codes related to this keyboard.

Modular Compact Alphanumeric Point of Sale Keyboard Layout

Figure 48 on page 288 shows the layout and assigned key-switch numbers for the Modular Compact Alphanumeric Point of Sale Keyboard.

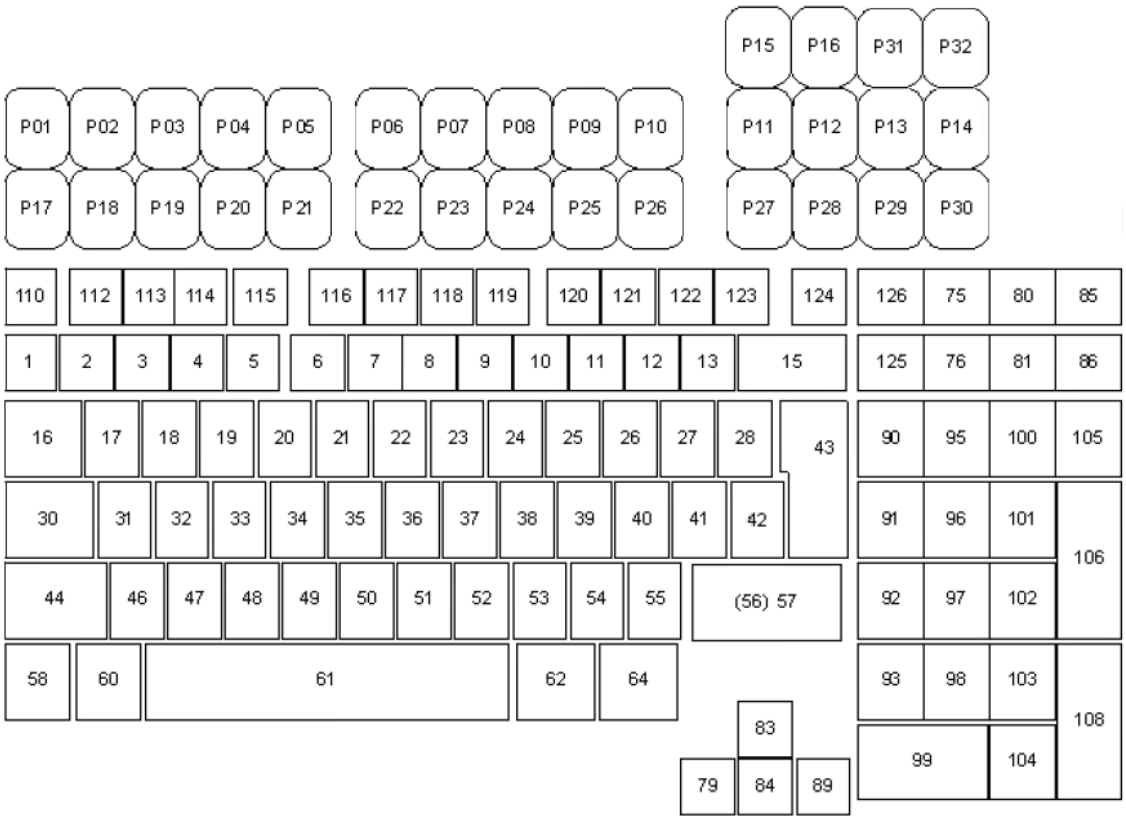


Figure 48. Modular CANPOS Keyboard Layout and Assigned Switch Numbers

Note: See Table 228 on page 296 for scan codes related to this keyboard.

PS/2 or USB (System Attached) scan code set for the Retail Alphanumeric Point of Sale Keyboard (NANPOS), Modular Alphanumeric Point of Sale Keyboard and the PC Point of Sale Keyboard (ANKPOS)

Table 227 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 227. PS/2 or USB (system attached) scan codes

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'29'	X'A9'	X'29'	
2	X'02'	X'82'	X'02'	
3	X'03'	X'83'	X'03'	
4	X'04'	X'84'	X'04'	
5	X'05'	X'85'	X'05'	
6	X'06'	X'86'	X'06'	
7	X'07'	X'87'	X'07'	
8	X'08'	X'88'	X'08'	
9	X'09'	X'89'	X'09'	
10	X'0a'	X'8a'	X'0a'	
11	X'0b'	X'8b'	X'0b'	
12	X'0c'	X'8c'	X'0c'	
13	X'0d'	X'8d'	X'0d'	
14	X'7d'	X'fd'	X'7d'	ANKPOS only
15	X'0e'	X'8e'	X'0e'	
16	X'0f'	X'8f'	X'0f'	
17	X'10'	X'90'	X'10'	
18	X'11'	X'91'	X'11'	
19	X'12'	X'92'	X'12'	
20	X'13'	X'93'	X'13'	
21	X'14'	X'94'	X'14'	
22	X'15'	X'95'	X'15'	
23	X'16'	X'96'	X'16'	
24	X'17'	X'97'	X'17'	
25	X'18'	X'98'	X'18'	
26	X'19'	X'99'	X'19'	
27	X'1a'	X'9a'	X'1a'	
28	X'1b'	X'9b'	X'1b'	
30	X'3a'	X'ba'	X'3a'	
31	X'1e'	X'9e'	X'1e'	
32	X'1f'	X'9f'	X'1f'	
33	X'20'	X'a0'	X'20'	
34	X'21'	X'a1'	X'21'	

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
35	X'22'	X'a2'	X'22'	
36	X'23'	X'a3'	X'23'	
37	X'24'	X'a4'	X'24'	
38	X'25'	X'a5'	X'25'	
39	X'26'	X'a6'	X'26'	
40	X'27'	X'a7'	X'27'	
41	X'28'	X'a8'	X'28'	
42	X'2b'	X'ab'	X'2b'	
43	X'1c'	X'9c'	X'1c'	
44	X'2a'	X'aa'	X'2a'	
45	X'56'	X'd6'	X'56'	World Trade key
46	X'2c'	X'ac'	X'2c'	
47	X'2d'	X'ad'	X'2d'	
48	X'2e'	X'ae'	X'2e'	
49	X'2f'	X'af'	X'2f'	
50	X'30'	X'b0'	X'30'	
51	X'31'	X'b1'	X'31'	
52	X'32'	X'b2'	X'32'	
53	X'33'	X'b3'	X'33'	
54	X'34'	X'b4'	X'34'	
55	X'35'	X'b5'	X'35'	
56	X'73'	X'f3'	X'73'	ANKPOS only
57	X'36'	X'b6'	X'36'	
58	X'1d'	X'9d'	X'1d'	
60	X'38'	X'b8'	X'38'	
61	X'39'	X'b9'	X'39'	
62	X'e0' X'38'	X'e0' X'b8'	X'01' X'38'	
64	X'e0' X'1d'	X'e0' X'9d'	X'01' X'1d'	
75	X'e0' X'52'	X'e0' X'd2'	X'01' X'52'	
76	X'e0' X'53'	X'e0' X'd3'	X'01' X'53'	
77	X'6a'	X'ea'	X'6a'	
78	X'6b'	X'eb'	X'6b'	
79	X'e0' X'4b'	X'e0' X'cb'	X'01' X'4b'	
80	X'e0' X'47'	X'e0' X'c7'	X'01' X'47'	
81	X'e0' X'4f'	X'e0' X'cf'	X'01' X'4f'	
82	X'6c'	X'ec'	X'6c'	
83	X'e0' X'48'	X'e0' X'c8'	X'01' X'48'	
84	X'e0' X'50'	X'e0' X'd0'	X'01' X'50'	
85	X'e0' X'49'	X'e0' X'c9'	X'01' X'49'	

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
86	X'e0' X'51'	X'e0' X'd1'	X'01' X'51'	
87	X'6d'	X'ed'	X'6d'	
88	X'6e'	X'ee'	X'6e'	
89	X'e0' X'4d'	X'e0' X'cd'	X'01' X'4d'	
90	X'6f'	X'ef'	X'6f'	
91	X'47'	X'c7'	X'47'	
92	X'4b'	X'cb'	X'4b'	
93	X'4f'	X'cf'	X'4f'	
94	X'52'	X'd2'	X'52'	
95	X'78'	X'f8'	X'78'	
96	X'48'	X'c8'	X'48'	
97	X'4c'	X'cc'	X'4c'	
98	X'50'	X'd0'	X'50'	
99	X'77'	X'f7'	X'52'	
100	X'65'	X'e5'	X'65'	
101	X'49'	X'c9'	X'49'	
102	X'4d'	X'cd'	X'4d'	
103	X'51'	X'd1'	X'51'	
104	X'53'	X'd3'	X'53'	
105	X'7a'	X'fa'	X'7a'	
106	X'7e'	X'fe'	X'7e'	
107	X'5f'	X'df'	X'5f'	When using a USB system keyboard with the OPOS driver the make code will be reported as X'66', not X'5f'.
108	X'71'	X'f1'	X'71'	
109	X'e0' X'1c'	X'e0' X'9c'	X'01' X'1c'	
110	X'01'	X'81'	X'01'	
112	X'3b'	X'bb'	X'3b'	
113	X'3c'	X'bc'	X'3c'	
114	X'3d'	X'bd'	X'3d'	
115	X'3e'	X'be'	X'3e'	
116	X'3f'	X'bf'	X'3f'	
117	X'40'	X'c0'	X'40'	
118	X'41'	X'c1'	X'41'	
119	X'42'	X'c2'	X'42'	
120	X'43'	X'c3'	X'43'	
121	X'44'	X'c4'	X'44'	
122	X'57'	X'd7'	X'57'	

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
123	X'58'	X'd8'	X'58'	
124	X'63'	X'e3'	X'63'	
125	X'74'	X'f4'	X'74'	
126	X'75'	X'f5'	X'75'	
127	X'76'	X'f6'	X'76'	
128	X'59'	X'd9'	X'59'	
131	X'7b'	X'fb'	X'7b'	ANKPOS only
132	X'79'	X'f9'	X'79'	ANKPOS only
133	X'70'	X'f0'	X'70'	ANKPOS only
135	X'72'	X'f2'	X'72'	
136	X'e0' X'2a' X'e0' X'37'	X'e0' X'b7' X'e0' X'aa'	X'01' X'37'	On Windows, key release only
137	X'46'	X'c6'	X'46'	
138	X'e1' X'1d' X'45' X'e1' X'9d' X'c5'		X'01'X'45'	Key press only
139	X'45'	X'c5'	X'45'	
	+ Alt Case			
136	X'54'	X'd4'	X'54'	On Windows, key release only
	+ Ctrl Case Pressed			
136	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
138	X'e0' X'46' X'e0' X'c6'		none	
	+ Left Ctrl Case Pressed			
124	X'9d' X'1c' X'1d'		X'02' X'1c'	Key press only
135	X'9d' X'01' X'1d'		X'02' X'01'	Key press only
	+ Right Ctrl Case Pressed			
124	X'e0' X'9d' X'1c' X'e0' X'1d'		X'02' X'1c'	Key press only
135	X'e0' X'9d' X'01' X'e0' X'1d'		X'02'X'01'	Key press only
	+ Both Ctrl Case Pressed			
124	X'e0' X'9d' X'9d' X'1c' X'e0' X'1d'X'e0'X'1d'		X'02' X'1c'	Key press only
135	X'e0' X'9d' X'9d' X'01' X'e0' X'1d'X'e0'X'1d'		X'02'X'01'	Key press only

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
	+ Shift Case Pressed			
136	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Left Shift Case Pressed, NUMLOCK OFF			
75	X'e0' X'aa' X'e0' X'52'	X'e0' X'd2' X'e0' X'2a'	X'01' X'52'	
76	X'e0' X'aa' X'e0' X'53'	X'e0' X'd3' X'e0' X'2a'	X'01' X'53'	
79	X'e0' X'aa' X'e0' X'4b'	X'e0' X'cb' X'e0' X'2a'	X'01' X'4b'	
80	X'e0' X'aa' X'e0' X'47'	X'e0' X'c7' X'e0' X'2a'	X'01' X'47'	
81	X'e0' X'aa' X'e0' X'4f'	X'e0' X'cf' X'e0' X'2a'	X'01' X'4f'	
83	X'e0' X'aa' X'e0' X'48'	X'e0' X'c8' X'e0' X'2a'	X'01' X'48'	
84	X'e0' X'aa' X'e0' X'50'	X'e0' X'd0' X'e0' X'2a'	X'01' X'50'	
85	X'e0' X'aa' X'e0' X'49'	X'e0' X'c9' X'e0' X'2a'	X'01' X'49'	
86	X'e0' X'aa' X'e0' X'51'	X'e0' X'd1' X'e0' X'2a'	X'01' X'51'	
89	X'e0' X'aa' X'e0' X'4d'	X'e0' X'cd' X'e0' X'2a'	X'01' X'4d'	
	+ Right Shift Case Pressed NUMLOCK OFF			
75	X'e0' X'b6' X'e0' X'52'	X'e0' X'd2' X'e0' X'36'	X'01' X'52'	
76	X'e0' X'b6' X'e0' X'53'	X'e0' X'd3' X'e0' X'36'	X'01' X'53'	
79	X'e0' X'b6' X'e0' X'4b'	X'e0' X'cb' X'e0' X'36'	X'01' X'4b'	
80	X'e0' X'b6' X'e0' X'47'	X'e0' X'c7' X'e0' X'36'	X'01' X'47'	
81	X'e0' X'b6' X'e0' X'4f'	X'e0' X'cf' X'e0' X'36'	X'01' X'4f'	
83	X'e0' X'b6' X'e0' X'48'	X'e0' X'c8' X'e0' X'36'	X'01' X'48'	
84	X'e0' X'b6' X'e0' X'50'	X'e0' X'd0' X'e0' X'36'	X'01' X'50'	
85	X'e0' X'b6' X'e0' X'49'	X'e0' X'c9' X'e0' X'36'	X'01' X'49'	

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
86	X'e0' X'b6' X'e0' X'51'	X'e0' X'd1' X'e0' X'36'	X'01' X'51'	
89	X'e0' X'b6' X'e0' X'4d'	X'e0' X'cd' X'e0' X'36'	X'01' X'4d'	
	+ Both Shift Case Pressed NUMLOCK OFF			
75	X'e0' X'aa' X'b6' X'e0' X'52'	X'e0' X'd2' X'e0' X'2a' X'e0' X'36'	X'01' X'52'	
76	X'e0' X'aa' X'b6' X'e0' X'53'	X'e0' X'd3' X'e0' X'2a' X'e0' X'36'	X'01' X'53'	
79	X'e0' X'aa' X'b6' X'e0' X'4b'	X'e0' X'cb' X'e0' X'2a' X'e0' X'36'	X'01' X'4b'	
81	X'e0' X'aa' X'b6' X'e0' X'4f'	X'e0' X'cf' X'e0' X'2a' X'e0' X'36'	X'01' X'4f'	
83	X'e0' X'aa' X'b6' X'e0' X'48'	X'e0' X'c8' X'e0' X'2a' X'e0' X'36'	X'01' X'48'	
84	X'e0' X'aa' X'b6' X'e0' X'50'	X'e0' X'd0' X'e0' X'2a' X'e0' X'36'	X'01' X'50'	
85	X'e0' X'aa' X'b6' X'e0' X'49'	X'e0' X'c9' X'e0' X'2a' X'e0' X'36'	X'01' X'49'	
86	X'e0' X'aa' X'b6' X'e0' X'51'	X'e0' X'd1' X'e0' X'2a' X'e0' X'36'	X'01' X'51'	
89	X'e0' X'aa' X'b6' X'e0' X'4d'	X'e0' X'cd' X'e0' X'2a' X'e0' X'36'	X'01' X'4d'	
	+ Shift + NUM LOCK ON			
75	X'e0' X'52'	X'e0' X'd2'	X'01' X'52'	
76	X'e0' X'53'	X'e0' X'd3'	X'01' X'53'	
79	X'e0' X'4b'	X'e0' X'cb'	X'01' X'4b'	
80	X'e0' X'47'	X'e0' X'c7'	X'01' X'47'	
81	X'e0' X'4f'	X'e0' X'cf'	X'01' X'4f'	
83	X'e0' X'48'	X'e0' X'c8'	X'01' X'48'	
84	X'e0' X'50'	X'e0' X'd0'	X'01' X'50'	
85	X'e0' X'49'	X'e0' X'c9'	X'01' X'49'	
86	X'e0' X'51'	X'e0' X'd1'	X'01' X'51'	
89	X'e0' X'4d'	X'e0' X'cd'	X'01' X'4d'	
	+ NUM LOCK ON			
75	X'e0' X'2a' X'e0' X'52'	X'e0' X'd2' X'e0' X'aa'	X'01' X'52'	
76	X'e0' X'2a' X'e0' X'53'	X'e0' X'd3' X'e0' X'aa'	X'01' X'53'	
79	X'e0' X'2a' X'e0' X'4b'	X'e0' X'cb' X'e0' X'aa'	X'01' X'4b'	

Table 227. PS/2 or USB (system attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
80	X'e0' X'2a' X'e0' X'47'	X'e0' X'c7' X'e0' X'aa'	X'01' X'47'	
81	X'e0' X'2a' X'e0' X'4f'	X'e0' X'cf' X'e0' X'aa'	X'01' X'4f'	
83	X'e0' X'2a' X'e0' X'48'	X'e0' X'c8' X'e0' X'aa'	X'01' X'48'	
84	X'e0' X'2a' X'e0' X'50'	X'e0' X'd0' X'e0' X'aa'	X'01' X'50'	
85	X'e0' X'2a' X'e0' X'49'	X'e0' X'c9' X'e0' X'aa'	X'01' X'49'	
86	X'e0' X'2a' X'e0' X'51'	X'e0' X'd1' X'e0' X'aa'	X'01' X'51'	
89	X'e0' X'2a' X'e0' X'4d'	X'e0' X'cd' X'e0' X'aa'	X'01' X'4d'	

Notes:

ANKPOS
PC Point of Sale Keyboard unique keys. These keys are present only on the PC Point of Sale (ANKPOS keyboard).

World Trade key
This key is present on all non-U.S. versions of the NANPOS keyboard, but not on the ANKPOS keyboard.

Double keys
Key switch numbers 94 and 99 are defined as double keys.

PS/2 scan code set for the Compact Alphanumeric Point of Sale Keyboard (CANPOS) and PS/2 / USB (System Attached) Modular Compact Alphanumeric Point of Sale Keyboard

Table 228 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 228. PS/2 or USB scan codes for CANPOS keyboard

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'29'	X'A9'	X'29'	
2	X'02'	X'82'	X'02'	
3	X'03'	X'83'	X'03'	
4	X'04'	X'84'	X'04'	
5	X'05'	X'85'	X'05'	
6	X'06'	X'86'	X'06'	
7	X'07'	X'87'	X'07'	
8	X'08'	X'88'	X'08'	
9	X'09'	X'89'	X'09'	
10	X'0a'	X'8a'	X'0a'	
11	X'0b'	X'8b'	X'0b'	
12	X'0c'	X'8c'	X'0c'	
13	X'0d'	X'8d'	X'0d'	
15	X'0e'	X'8e'	X'0e'	
16	X'0f'	X'8f'	X'0f'	
17	X'10'	X'90'	X'10'	
18	X'11'	X'91'	X'11'	
19	X'12'	X'92'	X'12'	
20	X'13'	X'93'	X'13'	
21	X'14'	X'94'	X'14'	
22	X'15'	X'95'	X'15'	
23	X'16'	X'96'	X'16'	
24	X'17'	X'97'	X'17'	
25	X'18'	X'98'	X'18'	
26	X'19'	X'99'	X'19'	
27	X'1a'	X'9a'	X'1a'	
28	X'1b'	X'9b'	X'1b'	
29	X'2b'	X'ab'	X'2b'	
30	X'3a'	X'ba'	X'3a'	
31	X'1e'	X'9e'	X'1e'	
32	X'1f'	X'9f'	X'1f'	
33	X'20'	X'a0'	X'20'	
34	X'21'	X'a1'	X'21'	
35	X'22'	X'a2'	X'22'	

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
36	X'23'	X'a3'	X'23'	
37	X'24'	X'a4'	X'24'	
38	X'25'	X'a5'	X'25'	
39	X'26'	X'a6'	X'26'	
40	X'27'	X'a7'	X'27'	
41	X'28'	X'a8'	X'28'	
42	X'2b'	X'ab'	X'2b'	
43	X'1c'	X'9c'	X'1c'	
44	X'2a'	X'aa'	X'2a'	
45	X'56'	X'd6'	X'56'	
46	X'2c'	X'ac'	X'2c'	
47	X'2d'	X'ad'	X'2d'	
48	X'2e'	X'ae'	X'2e'	
49	X'2f'	X'af'	X'2f'	
50	X'30'	X'b0'	X'30'	
51	X'31'	X'b1'	X'31'	
52	X'32'	X'b2'	X'32'	
53	X'33'	X'b3'	X'33'	
54	X'34'	X'b4'	X'34'	
55	X'35'	X'b5'	X'35'	
56	X'73'	X'f3'	X'73'	
57	X'36'	X'b6'	X'36'	
58	X'1d'	X'9d'	X'1d'	
60	X'38'	X'b8'	X'38'	
61	X'39'	X'b9'	X'39'	
62	X'e0' X'38'	X'e0' X'b8'	X'01' X'38'	
64	X'e0' X'1d'	X'e0' X'9d'	X'01' X'1d'	
75	X'e0' X'52'	X'e0' X'd2'	X'01' X'52'	
76	X'e0' X'53'	X'e0' X'd3'	X'01' X'53'	
79	X'e0' X'4b'	X'e0' X'cb'	X'01' X'4b'	
80	X'e0' X'47'	X'e0' X'c7'	X'01' X'47'	
81	X'e0' X'4f'	X'e0' X'cf'	X'01' X'4f'	
83	X'e0' X'48'	X'e0' X'c8'	X'01' X'48'	
84	X'e0' X'50'	X'e0' X'd0'	X'01' X'50'	
85	X'e0' X'49'	X'e0' X'c9'	X'01' X'49'	
86	X'e0' X'51'	X'e0' X'd1'	X'01' X'51'	
89	X'e0' X'4d'	X'e0' X'cd'	X'01' X'4d'	
90	X'45'	X'c5'	X'45'	
91	X'47'	X'c7'	X'47'	

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
92	X'4b'	X'cb'	X'4b'	
93	X'4f'	X'cf'	X'4f'	
95	X'e0' X'35'	X'e0' X'b5'	X'01' X'35'	See 302.
96	X'48'	X'c8'	X'48'	
97	X'4c'	X'cc'	X'4c'	
98	X'50'	X'd0'	X'50'	
99	X'52'	X'd2'	X'52'	
100	X'37'	X'b7'	X'37'	See 302.
101	X'49'	X'c9'	X'49'	
102	X'4d'	X'cd'	X'4d'	
103	X'51'	X'd1'	X'51'	
104	X'53'	X'd3'	X'53'	See 302.
105	X'4a'	X'ca'	X'4a'	See 302.
106	X'4e'	X'ce'	X'4e'	See 302.
107	X'7e'	X'fe'	X'7e'	
108	X'e0' X'1c'	X'e0' X'9c'	X'01' X'1c'	
109	X'78'	X'f8'	X'78'	
110	X'01'	X'81'	X'01'	
112	X'3b'	X'bb'	X'3b'	
113	X'3c'	X'bc'	X'3c'	
114	X'3d'	X'bd'	X'3d'	
115	X'3e'	X'be'	X'3e'	
116	X'3f'	X'bf'	X'3f'	
117	X'40'	X'c0'	X'40'	
118	X'41'	X'c1'	X'41'	
119	X'42'	X'c2'	X'42'	
120	X'43'	X'c3'	X'43'	
121	X'44'	X'c4'	X'44'	
122	X'57'	X'd7'	X'57'	
123	X'58'	X'd8'	X'58'	
124	X'e0' X'2a' X'e0' X'37'	X'e0' X'b7' X'e0' X'aa'	X'01' X'37'	On Windows, key release only
125	X'46'	X'c6'	X'46'	
126	X'e1' X'1d' X'45' X'e1' X'9d' X'c5'		X'01'X'45'	Key press only
	+ Alt Case			
124	X'54'	X'd4'	X'54'	On Windows, key release only
	+ Left or Right Ctrl Pressed			

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
126	X'e0' X'46' X'e0' X'c6'		none	On Windows, key release only
	+ Left Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Right Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Both Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Left Shift Case Pressed			
75	X'e0' X'aa' X'e0' X'52'	X'e0' X'd2' X'e0' X'2a'	X'01' X'52'	
76	X'e0' X'aa' X'e0' X'53'	X'e0' X'd3' X'e0' X'2a'	X'01' X'53'	
79	X'e0' X'aa' X'e0' X'4b'	X'e0' X'cb' X'e0' X'2a'	X'01' X'4b'	
80	X'e0' X'aa' X'e0' X'47'	X'e0' X'c7' X'e0' X'2a'	X'01' X'47'	
81	X'e0' X'aa' X'e0' X'4f'	X'e0' X'cf' X'e0' X'2a'	X'01' X'4f'	
83	X'e0' X'aa' X'e0' X'48'	X'e0' X'c8' X'e0' X'2a'	X'01' X'48'	
84	X'e0' X'aa' X'e0' X'50'	X'e0' X'd0' X'e0' X'2a'	X'01' X'50'	
85	X'e0' X'aa' X'e0' X'49'	X'e0' X'c9' X'e0' X'2a'	X'01' X'49'	
86	X'e0' X'aa' X'e0' X'51'	X'e0' X'd1' X'e0' X'2a'	X'01' X'51'	
89	X'e0' X'aa' X'e0' X'4d'	X'e0' X'cd' X'e0' X'2a'	X'01' X'4d'	
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Right Shift Case Pressed			
75	X'e0' X'b6' X'e0' X'52'	X'e0' X'd2' X'e0' X'36'	X'01' X'52'	
76	X'e0' X'b6' X'e0' X'53'	X'e0' X'd3' X'e0' X'36'	X'01' X'53'	

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
79	X'e0' X'b6' X'e0' X'4b'	X'e0' X'cb' X'e0' X'36'	X'01' X'4b'	
80	X'e0' X'b6' X'e0' X'47'	X'e0' X'c7' X'e0' X'36'	X'01' X'47'	
81	X'e0' X'b6' X'e0' X'4f'	X'e0' X'cf' X'e0' X'36'	X'01' X'4f'	
83	X'e0' X'b6' X'e0' X'48'	X'e0' X'c8' X'e0' X'36'	X'01' X'48'	
84	X'e0' X'b6' X'e0' X'50'	X'e0' X'd0' X'e0' X'36'	X'01' X'50'	
85	X'e0' X'b6' X'e0' X'49'	X'e0' X'c9' X'e0' X'36'	X'01' X'49'	
86	X'e0' X'b6' X'e0' X'51'	X'e0' X'd1' X'e0' X'36'	X'01' X'51'	
89	X'e0' X'b6' X'e0' X'4d'	X'e0' X'cd' X'e0' X'36'	X'01' X'4d'	
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Both Shift Case Pressed			
75	X'e0' X'aa' X'b6' X'e0' X'52'	X'e0' X'd2' X'e0' X'2a' X'e0' X'36'	X'01' X'52'	
76	X'e0' X'aa' X'b6' X'e0' X'53'	X'e0' X'd3' X'e0' X'2a' X'e0' X'36'	X'01' X'53'	
79	X'e0' X'aa' X'b6' X'e0' X'4b'	X'e0' X'cb' X'e0' X'2a' X'e0' X'36'	X'01' X'4b'	
81	X'e0' X'aa' X'b6' X'e0' X'4f'	X'e0' X'cf' X'e0' X'2a' X'e0' X'36'	X'01' X'4f'	
83	X'e0' X'aa' X'b6' X'e0' X'48'	X'e0' X'c8' X'e0' X'2a' X'e0' X'36'	X'01' X'48'	
84	X'e0' X'aa' X'b6' X'e0' X'50'	X'e0' X'd0' X'e0' X'2a' X'e0' X'36'	X'01' X'50'	
85	X'e0' X'aa' X'b6' X'e0' X'49'	X'e0' X'c9' X'e0' X'2a' X'e0' X'36'	X'01' X'49'	
86	X'e0' X'aa' X'b6' X'e0' X'51'	X'e0' X'd1' X'e0' X'2a' X'e0' X'36'	X'01' X'51'	
89	X'e0' X'aa' X'b6' X'e0' X'4d'	X'e0' X'cd' X'e0' X'2a' X'e0' X'36'	X'01' X'4d'	
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	On Windows, key release only
	+ Shift + NUM LOCK ON			
75	X'e0' X'52'	X'e0' X'd2'	X'01' X'52'	
76	X'e0' X'53'	X'e0' X'd3'	X'01' X'53'	
79	X'e0' X'4b'	X'e0' X'cb'	X'01' X'4b'	
80	X'e0' X'47'	X'e0' X'c7'	X'01' X'47'	

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
81	X'e0' X'4f'	X'e0' X'cf'	X'01' X'4f'	
83	X'e0' X'48'	X'e0' X'c8'	X'01' X'48'	
84	X'e0' X'50'	X'e0' X'd0'	X'01' X'50'	
85	X'e0' X'49'	X'e0' X'c9'	X'01' X'49'	
86	X'e0' X'51'	X'e0' X'd1'	X'01' X'51'	
89	X'e0' X'4d'	X'e0' X'cd'	X'01' X'4d'	
	+ NUM LOCK ON			
75	X'e0' X'2a' X'e0' X'52'	X'e0' X'd2' X'e0' X'aa'	X'01' X'52'	
76	X'e0' X'2a' X'e0' X'53'	X'e0' X'd3' X'e0' X'aa'	X'01' X'53'	
79	X'e0' X'2a' X'e0' X'4b'	X'e0' X'cb' X'e0' X'aa'	X'01' X'4b'	
80	X'e0' X'2a' X'e0' X'47'	X'e0' X'c7' X'e0' X'aa'	X'01' X'47'	
81	X'e0' X'2a' X'e0' X'4f'	X'e0' X'cf' X'e0' X'aa'	X'01' X'4f'	
83	X'e0' X'2a' X'e0' X'48'	X'e0' X'c8' X'e0' X'aa'	X'01' X'48'	
84	X'e0' X'2a' X'e0' X'50'	X'e0' X'd0' X'e0' X'aa'	X'01' X'50'	
85	X'e0' X'2a' X'e0' X'49'	X'e0' X'c9' X'e0' X'aa'	X'01' X'49'	
86	X'e0' X'2a' X'e0' X'51'	X'e0' X'd1' X'e0' X'aa'	X'01' X'51'	
89	X'e0' X'2a' X'e0' X'4d'	X'e0' X'cd' X'e0' X'aa'	X'01' X'4d'	
	P01 - P32			
P01	X'e0' X'0b'	X'e0' X'8b'	X'01' X'0b'	
P02	X'e0' X'02'	X'e0' X'82'	X'01' X'02'	
P03	X'e0' X'03'	X'e0' X'83'	X'01' X'03'	
P04	X'e0' X'04'	X'e0' X'84'	X'01' X'04'	
P05	X'e0' X'05'	X'e0' X'85'	X'01' X'05'	
P06	X'e0' X'06'	X'e0' X'86'	X'01' X'06'	
P07	X'e0' X'07'	X'e0' X'87'	X'01' X'07'	
P08	X'e0' X'08'	X'e0' X'88'	X'01' X'08'	
P09	X'e0' X'09'	X'e0' X'89'	X'01' X'09'	
P10	X'e0' X'0a'	X'e0' X'8a'	X'01' X'0a'	
P11	X'e0' X'1e'	X'e0' X'9e'	X'01' X'1e'	
P12	X'e0' X'30'	X'e0' X'b0'	X'01' X'30'	
P13	X'e0' X'2e'	X'e0' X'ae'	X'01' X'2e'	
P14	X'e0' X'20'	X'e0' X'a0'	X'01' X'20'	

Table 228. PS/2 or USB scan codes for CANPOS keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
P15	X'e0' X'12'	X'e0' X'92'	X'01' X'12'	
P16	X'e0' X'21'	X'e0' X'a1'	X'01' X'21'	
P17	X'e0' X'22'	X'e0' X'a2'	X'01' X'22'	
P18	X'e0' X'23'	X'e0' X'a3'	X'01' X'23'	
P19	X'e0' X'17'	X'e0' X'97'	X'01' X'17'	
P20	X'e0' X'24'	X'e0' X'a4'	X'01' X'24'	
P21	X'e0' X'25'	X'e0' X'a5'	X'01' X'25'	
P22	X'e0' X'26'	X'e0' X'a6'	X'01' X'26'	
P23	X'e0' X'32'	X'e0' X'b2'	X'01' X'32'	
P24	X'e0' X'31'	X'e0' X'b1'	X'01' X'31'	
P25	X'e0' X'18'	X'e0' X'98'	X'01' X'18'	
P26	X'e0' X'19'	X'e0' X'99'	X'01' X'19'	
P27	X'e0' X'10'	X'e0' X'90'	X'01' X'10'	
P28	X'e0' X'13'	X'e0' X'93'	X'01' X'13'	
P29	X'e0' X'1f'	X'e0' X'9f'	X'01' X'1f'	
P30	X'e0' X'14'	X'e0' X'94'	X'01' X'14'	
P31	X'e0' X'16'	X'e0' X'96'	X'01' X'16'	
P32	X'e0' X'2f'	X'e0' X'af'	X'01' X'2f'	

Note: Linux: When connected as USB System, the Modular CANPOS Keyboard will return the following JavaPOS POS key data for the following keys:

Key 95 - X'53'
Key 100 - X'65'
Key 104 - X'78'"
Key 105 - X'7a'"
Key 106 - X'7e'

Point of Sale Keyboard V

Figure 49 shows the Keyboard V layout.

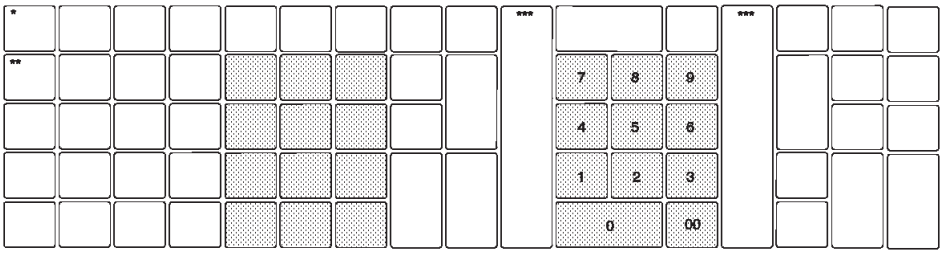


Figure 49. Point of Sale Keyboard V layout

Notes:

1. The single asterisk (*) in the top-left portion of the keyboard indicates the S1 key.
2. The double asterisk (**) in the top-left portion of the keyboard indicates the S2 key.
3. The triple asterisk (***) indicates that a 1×5 dummy cap covers those key switches.
4. The two possible locations for the numeric keypad are shown in the shaded area of the illustration. The default location for the numeric keypad is the right-most shaded area.

Keyboard V scan codes

Figure 50 shows the key scan codes for the Keyboard-V.

4C {00}	4F	4E	4D	49	48	47	4B	4A	(46)	45	(44)	43	(42)	41	40	50
3C {01}	3F	3E	3D	39	38	37	3B	(3A)	(36)	35	34	33	(32)	(31)	30	51
2C	2F	2E	2D	29	28	27	2B	2A	(26)	25	24	23	(22)	21	20	52
1C	1F	1E	1D	19	18	17	(1B)	(1A)	(16)	15	14	13	(12)	11	(10)	(53)
0C	0F	0E	0D	09	08	07	0B	0A	(06)	05	(04)	03	(02)	81	80	54

Figure 50. Keyboard-V scan code set

Notes:

1. ‘( )’ indicates that these scan codes can be generated when the layout is changed.
2. ‘{ }’ indicates these scan codes will be generated only when the keylock is in the **system** position.
3. Each double key produces only one scan code (the key scan code without parentheses in the illustration), unlike the single-byte character set (SBCS) keyboards.
4. Return only make scan codes.

PLU Keyboard and Display III

Figure 51 shows the PLU Keyboard and Display-III layout and scan codes.

The two possible locations for the numeric keypad are shaded in Figure 51. The default location for the numeric keypad is the right-most shaded area.

7F	7E	7D	79	78	77	7C	7B	7A	76	75	74	73	72	71	70
6F	6E	6D	69	68	67	6C	6B	6A	66	65	64	63	62	61	60
5F	5E	5D	59	58	57	5C	5B	5A	56	55	54	53	52	51	50
4F	4E	4D	49	48	47	4C	4B	4A	46	45	(44)	43	42	41	40
3F	3E	3D	39	38	37	3C	3B	3A	(36)	35	34	33	(32)	31	30
2F	2E	2D	29	28	27	2C	2B	2A	26	25	24	23	22	21	20
1F	1E	1D	19	18	17	1C	1B	(1A)	(16)	15	14	13	12	(11)	(10)
0F	0E	0D	09	08	07	0C	0B	0A	06	05	(04)	03	02	01	00

Figure 51. PLU Keyboard and Display-III layout

Notes:

1. “( )” indicates that these scan codes can be generated when the layout is changed.
2. Each double key produces only one scan code (the key scan code without parentheses in the illustration), unlike the single-byte character set keyboards.
3. Return only make scan codes.
4. Only double keys are shown in the figure.

4674 Point of Sale Keyboard (built-in)

Figure 52 shows the layout of the keyboard.

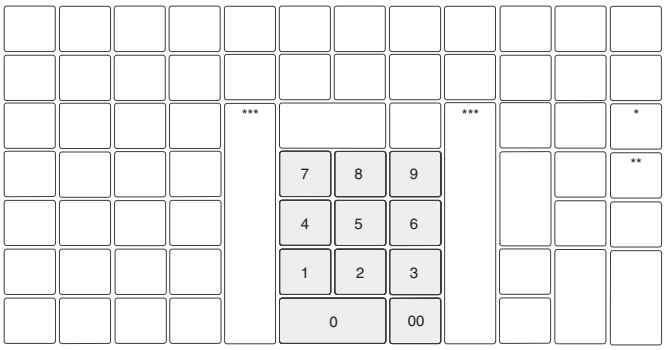


Figure 52. 4674 Point of Sale Keyboard (built-in)

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicates that a 1×5 dummy cap covers those key switches.

4674 POS Keyboard (built-in) scan codes

Figure 53 provides the scan codes.

6Ah	69h	68h	67h	66h	65h	64h	63h	62h	6Fh	6Eh	6Dh
5Ah	59h	58h	57h	56h	55h	54h	53h	52h	5Fh	5Eh	5Dh
4Ah	49h	48h	47h	46h	45h	44h	43h	42h	4Fh	4Eh	4Dh (00)
3Ah	39h	38h	37h	36h	35h	34h	33h	32h	3Fh	3Eh	3Dh (01)
2Ah	29h	28h	27h	26h	25h	24h	23h	22h	2Fh	2Eh	2Dh
1Ah	19h	18h	17h	16h	15h	14h	13h	12h	1Fh	1Eh	1Dh
0Ah	09h	08h	07h	06h	05h	04h	03h	02h	0Fh	0Eh	0Dh

Figure 53. Scan codes for 4674 built-in keyboard

xxh Indicates scan codes that are generated in the default keyboard configuration

( ) Indicates scan codes that are generated only when the keylock is in the **System** position.

Note: Return only make scan codes.

4685 keyboards

This section describes all of the 4685 keyboards.

All 4685 keyboards return only make scan codes.

4685 Point of Sale Keyboard Model K01

Figure 54 shows the layout of the 4685 Keyboard Model K01.

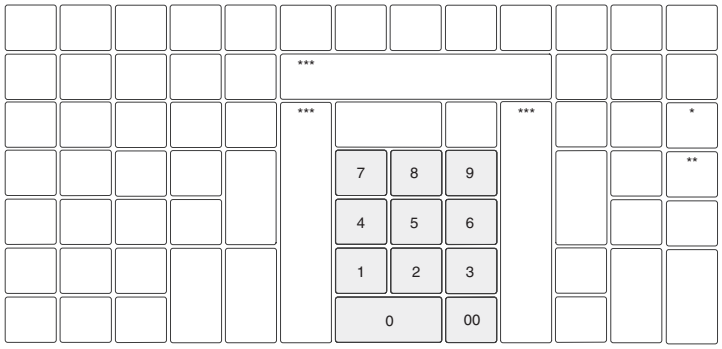


Figure 54. 4685 Point of Sale Keyboard Model K01 layout

Notes:

1. The numeric keypad location is shaded in the illustration.
2. The single asterisk (*) in the middle-right portion of the keyboard indicates the S1 key.
3. The double asterisk (**) in the middle-right portion of the keyboard indicates the S2 key.
4. The triple asterisk (***) indicates that a 1×5 dummy cap covers those key switches.

4685 Keyboard Model K01 scan code set

Figure 55 shows the key scan codes for the 4685 Keyboard Model K01.

6B	6A	69	68	67	66	65	64	63	62	6F	6E	6D
5B	5A	59	58	57	*** 56	55	54	53	52	5F	5E	5D
4B	4A	49	48	47	*** 46	45	44	43	*** 42	4F	4E	* 4D (00)
3B	3A	39	38	37	36	35	34	33	32	3F	3E	** 3D (01)
2B	2A	29	28	27	26	25	24	23	22	2F	2E	2D
1B	1A	19	18	17	16	15	14	13	12	1F	1E	1D
0B	0A	09	08	07	06	05	04	03	02	0F	0E	0D

Figure 55. 4685 Keyboard Model K01 scan code set

Notes:

1. “( )” indicates these scan codes will be generated only when the keylock is in the “system” position.
2. Each double key produces only one scan code.

3. The single asterisk (*) in the middle-right portion of the keyboard indicates the S1 key.
4. The double asterisk (**) in the middle-right portion of the keyboard indicates the S2 key.
5. The triple asterisk (***) indicates that a 1×5 dummy cap covers those key switches.

4685 Keyboard K02 Ultra 7 (four-position keylock)

Figure 56 shows the layout of the keyboard.

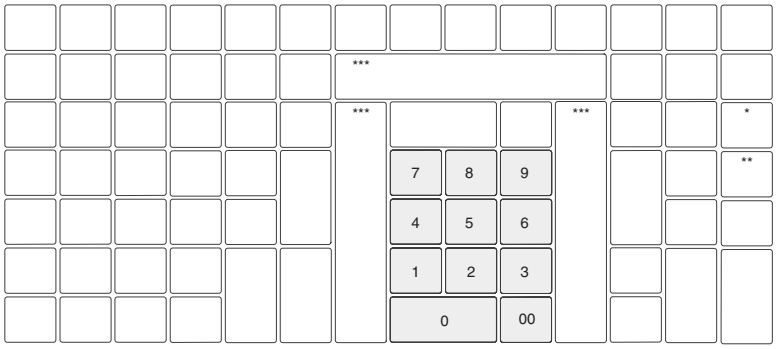


Figure 56. 4685 Keyboard K02 Ultra 7 with 4-position keylock

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicate dummy keys.

4685 Keyboard K02 Ultra 7 scan codes

Figure 57 shows the scan codes for the 4685 Keyboard K02 Ultra 7.

6Ch	6Bh	6Ah	69h	68h	67h	66h	65h	64h	63h	62h	6Fh	6Eh	6Dh
5Ch	5Bh	5Ah	59h	58h	57h	56h	55h	54h	53h	52h	5Fh	5Eh	5Dh
4Ch	4Bh	4Ah	49h	48h	47h	46h	45h	44h	43h	42h	4Fh	4Eh	4Dh (00)
3Ch	3Bh	3Ah	39h	38h	<37h>	36h	35h	34h	33h	32h	3Fh	3Eh	3Dh (01)
2Ch	2Bh	2Ah	29h	28h	27h	26h	25h	24h	23h	22h	2Fh	2Eh	2Dh
1Ch	1Bh	1Ah	19h	<18h>	<17h>	16h	15h	14h	13h	12h	1Fh	1Eh	1Dh
0Ch	0Bh	0Ah	09h	08h	07h	06h	05h	04h	03h	02h	0Fh	0Eh	0Dh

Figure 57. 4685 Keyboard K02 Ultra 7 scan codes

Notes:

1. < > indicates these are dummy keys at default
2. ( ) indicates scan codes that are generated only when the keylock is in the *System* position.

4685 Keyboard Model K02 Ultra 7 with MSR/E (four or six position keylock)

Figure 58 shows the layout of the keyboard.

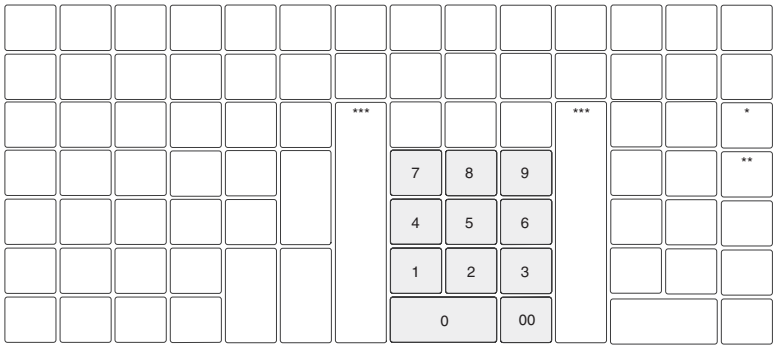


Figure 58. 4685 Keyboard K02 Ultra 7 with MSR/E (4 or 6 position keylock)

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicate dummy keys.

4685 Keyboard K02 Ultra 7 with MSR/E scan codes

Figure 59 provides the scan codes.

6Ch	6Bh	6Ah	69h	68h	67h	66h	65h	64h	63h	62h	6Fh	6Eh	6Dh
5Ch	5Bh	5Ah	59h	58h	57h	56h	55h	54h	53h	52h	5Fh	5Eh	5Dh
4Ch	4Bh	4Ah	49h	48h	47h	46h	45h	44h	43h	42h	4Fh	4Eh	4Dh (00)
3Ch	3Bh	3Ah	39h	38h	37h	36h	35h	34h	33h	32h	3Fh	3Eh	3Dh (01)
2Ch	2Bh	2Ah	29h	28h	27h	26h	25h	24h	23h	22h	2Fh	2Eh	2Dh
1Ch	1Bh	1Ah	19h	18h	17h	16h	15h	14h	13h	12h	1Fh	1Eh	1Dh
0Ch	0Bh	0Ah	09h	08h	07h	06h	05h	04h	03h	02h	0Fh	0Eh	0Dh

Figure 59. 4685 Keyboard K02 Ultra 7 with MSR/E scan codes

xxh Indicates scan codes that are generated in the default keyboard configuration

( ) Indicates scan codes that are generated only when the keylock is in the **System** position.

4685 Keyboard Model KC1

Figure 60 shows the keyboard layout.

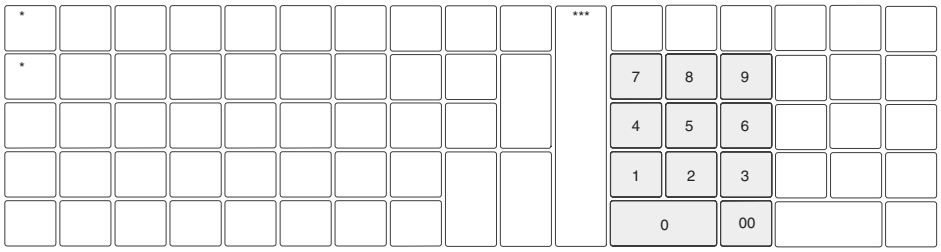


Figure 60. 4685 Model KC1

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicate dummy keys.

4685 Keyboard Model KC1 scan codes

Figure 61 provides the scan codes.

4Ch (00)	4Fh	4Eh	4Dh	49h	48h	47h	4Bh	4Ah	46h	45h	44h	43h	42h	41h	40h	50h
3Ch (00)	3Fh	3Eh	3Dh	39h	38h	37h	3Bh	3Ah	36h	35h	34h	33h	32h	31h	30h	51h
2Ch	2Fh	2Eh	2Dh	29h	28h	27h	2Bh	2Ah	26h	25h	24h	23h	22h	21h	20h	52h
1Ch	1Fh	1Eh	1Dh	19h	18h	17h	1Bh	1Ah	16h	15h	14h	13h	12h	11h	10h	53h
0Ch	0Fh	0Eh	0Dh	09h	08h	07h	0Bh	0Ah	06h	05h	04h	03h	02h	81h	80h	54h

Figure 61. Scan codes for Model 4685-KC1

xxh Indicates scan codes that are generated in the default keyboard configuration

( ) Indicates scan codes that are generated only when the keylock is in the *System* position.

4685 Keyboard Model K03

Figure 62 shows the layout of the keyboard.

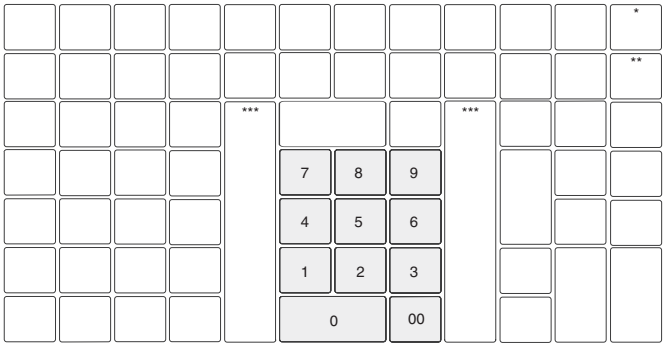


Figure 62. 4685 Keyboard Model K03 layout

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicate dummy keys.

4685 Keyboard Model K03 scan codes

Figure 63 provides the scan codes.

0x6A	0x69	0x68	0x67	0x66	0x65	0x64	0x63	0x62	0x6F	0x6E	0x6D (0x00)
0x5A	0x59	0x58	0x57	0x56	0x55	0x54	0x53	0x52	0x5F	0x5E	0x5D (0x01)
0x4A	0x49	0x48	0x47	<0x46>	0x45	<0x44>	0x43	<0x42>	0x4F	0x4E	0x4D
0x3A	0x39	0x38	0x37	<0x36>	0x35	0x34	0x33	<0x32>	<0x3F>	0x3E	0x3D
0x2A	0x29	0x28	0x27	<0x26>	0x25	0x24	0x23	<0x22>	0x2F	0x2E	0x2D
0x1A	0x19	0x18	0x17	<0x16>	0x15	0x14	0x13	<0x12>	<0x1F>	<0x1E>	<0x1D>
0x0A	0x09	0x08	0x07	<0x06>	0x05	0x04	0x03	<0x02>	0x0F	0x0E	0x0D

Figure 63. 4685 Keyboard Model K03 scan codes

Notes:

1. <> indicates these are dummy keys at default.
2. () indicates scan codes that are generated only when the keylock is in the *System* position.
3. RS-485 connectivity returns only make scan codes.

4820 IBM SurePoint Solution Keypad

Figure 64 shows the layout of the key-switch numbers for the 4820 Keypad.

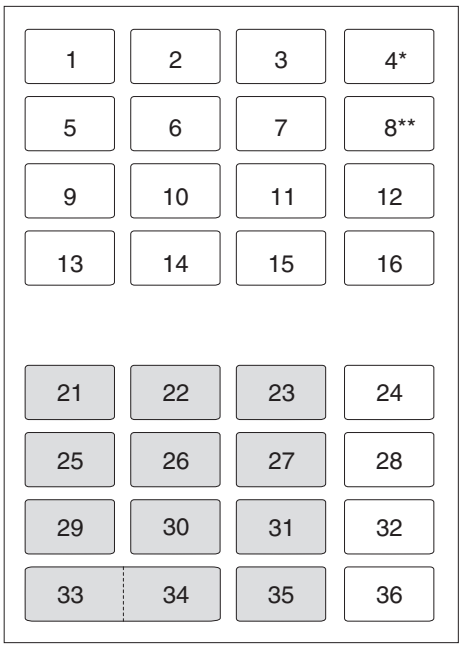


Figure 64. 4820 Keypad key-switch layout

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.

Table 229 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the RS-485 or USB port.

Table 229. 4820 RS-485/USB IBM Surepoint Solution keypad scan codes

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	X'4b'	X'f0' X'4b'	X'4b'	
2	X'3b'	X'f0' X'3b'	X'3b'	
3	X'6b'	X'f0' X'6b'	X'6b'	
4	X'8b'	X'f0' X'8b'	X'8b'	This is the S1 key
5	X'4c'	X'f0' X'4c'	X'4c'	
6	X'3c'	X'f0' X'3c'	X'3c'	
7	X'6c'	X'f0' X'6c'	X'6c'	
8	X'8c'	X'f0' X'8c'	X'8c'	This is the S2 key.
9	X'4f'	X'f0' X'4f'	X'4f'	
10	X'3f'	X'f0' X'3f'	X'3f'	
11	X'6f'	X'f0' X'6f'	X'6f'	
12	X'8f'	X'f0' X'8f'	X'8f'	
13	X'4e'	X'f0' X'4e'	X'4e'	

Table 229. 4820 RS-485/USB IBM Surepoint Solution keypad scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
14	X'3e'	X'f0' X'3e'	X'3e'	
15	X'6e'	X'f0' X'6e'	X'6e'	
16	X'8e'	X'f0' X'8e'	X'8e'	
21	X'7c'	X'f0' X'7c'	X'7c'	
22	X'0c'	X'f0' X'0c'	X'0c'	
23	X'1c'	X'f0' X'1c'	X'1c'	
24	X'50'	X'f0' X'50'	X'50'	
25	X'7f'	X'f0' X'7f'	X'7f'	
26	X'0f'	X'f0' X'0f'	X'0f'	
27	X'1f'	X'f0' X'1f'	X'1f'	
28	X'9f'	X'f0' X'9f'	X'9f'	
29	X'7e'	X'f0' X'7e'	X'7e'	
30	X'0e'	X'f0' X'0e'	X'0e'	
31	X'1e'	X'f0' X'1e'	X'1e'	
32	X'9e'	X'f0' X'9e'	X'9e'	
33	X'7d'	X'f0' X'7d'	X'7d'	
34	X'0d'	X'f0' X'0d'	X'0d'	
35	X'1d'	X'f0' X'1d'	X'1d'	
36	X'9d'	X'f0' X'9d'	X'9d'	
	+ Ctrl Case Pressed			
4	X'f0' X'50' X'00' X'50'		X'00'	Key press only
8	X'f0' X'50' X'01' X'50'		X'01'	Key press only

4820 IBM SurePoint Solution Keypad (system-attached)

Figure 65 shows the layout of the key-switch numbers for the 4820 Keypad.

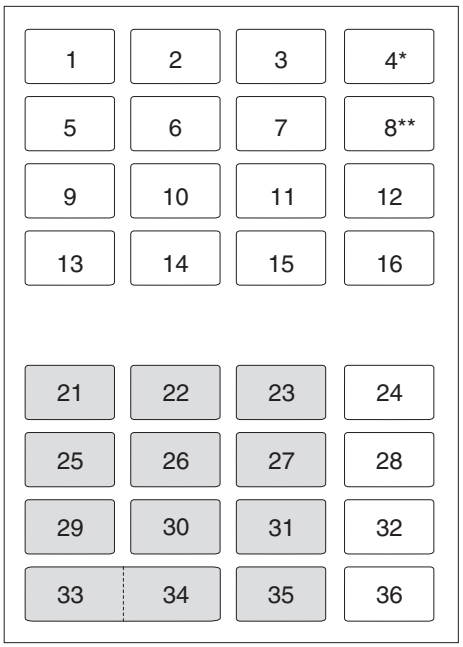


Figure 65. 4820 Keypad (system-attached) key-switch layout

Table 230 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 230. 4820 IBM SurePoint Solution Keypad (system-attached) scan codes

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData
1	X'3b'	X'bb'	X'3b'
2	X'3c'	X'bc'	X'3c'
3	X'3d'	X'bd'	X'3d'
4	X'3e'	X'be'	X'3e'
5	X'3f'	X'bf'	X'3f'
6	X'40'	X'c0'	X'40'
7	X'41'	X'c1'	X'41'
8	X'42'	X'c2'	X'42'
9	X'43'	X'c3'	X'43'
10	X'44'	X'c4'	X'44'
11	X'57'	X'd7'	X'57'
12	X'58'	X'd8'	X'58'
13	X'01'	X'81'	X'01'
14	X'4e'	X'ce'	X'4e'
15	X'4a'	X'ca'	X'4a'
16	X'0e'	X'8e'	X'0e'
21	X'47'	X'c7'	X'47'
22	X'48'	X'c8'	X'48'
23	X'49'	X'c9'	X'49'
24	X'1d'	X'9d'	X'1d'
25	X'4b'	X'cb'	X'4b'
26	X'4c'	X'cc'	X'4c'
27	X'4d'	X'cd'	X'4d'
28	X'38'	X'b8'	X'38'
29	X'4f'	X'cf'	X'4f'
30	X'50'	X'd0'	X'50'
31	X'51'	X'd1'	X'51'
32	X'39'	X'b9'	X'39'
33	X'52'	X'd2'	X'52'
34	X'e0' X'35'	X'e0' X'b5'	X'01' X'35'
35	X'37'	X'b7'	X'37'
36	X'e0' X'1c'	X'e0' X'9c'	X'01' X'1c'

SureOne model 4614/4615 and SurePOS 100 model 4613 keyboard (built-in)

SureOne supports only the 101-key layout.

SurePOS 100 supports 101-, 102-, and 103-key layouts.

Notes:

1. When the keyboard is configured to be in POS mode, the application program must use standard PC Keyboard interfaces to intercept the keystrokes (scan codes) and perform the appropriate translation and detection.
2. JavaPOS does not support POS mode.

Figure 66 shows the layout of the key-switch numbers for the SureOne/SurePOS 100 keyboards.

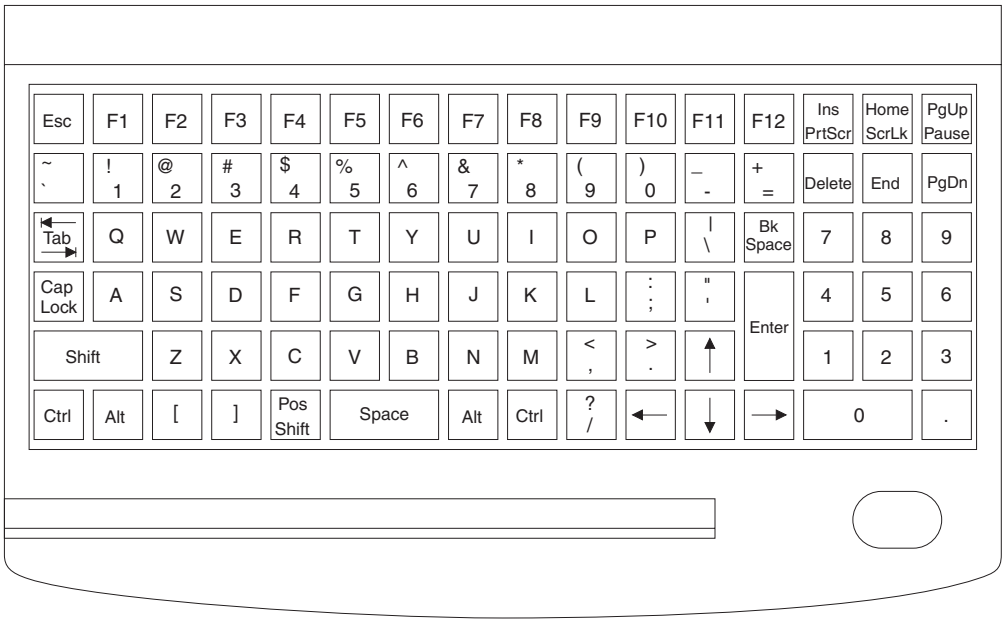


Figure 66. SureOne/SurePOS 100 keyboard (built-in) keyboard layout

SureOne model 4614/4615 and SurePOS 100 model 4613 keyboard (built-in) scan codes

Figure 67 provides the key switch numbers for the SureOne/SurePOS 100 101-key keyboard.

Key Switch Number

110	112	113	114	115	116	117	118	119	120	121	122	123	75	80	85
1	2	3	4	5	6	7	8	9	10	11	12	13	76	81	86
16	17	18	19	20	21	22	23	24	25	26	29	15	91	96	101
30	31	32	33	34	35	36	37	38	39	40	41	43	92	97	102
44		46	47	48	49	50	51	52	53	54	83		93	98	103
58	60	27	28	(*1)		61	62	64	55	79	84	89	99	94	104

Figure 67. SureOne/SurePOS 100 101-key keyboard switches

Figure 68 on page 317 provides the key switch numbers for the SurePOS 100 102-key keyboard.

Key Switch Number

110	112	113	114	115	116	117	118	119	120	121	122	123	75	80	85
1	2	3	4	5	6	7	8	9	10	11	12	13	76	81	86
16	17	18	19	20	21	22	23	24	25	26	42	15	91	96	101
30	31	32	33	34	35	36	37	38	39	40	41	43	92	97	102
44	45	46	47	48	49	50	51	52	53	54	83		93	98	103
58	60	27	28	(*1)		61	62	64	55	79	84	89	99	94	104

Figure 68. SurePOS 100 102-key keyboard switches

Figure 69 provides the key switch numbers for the SurePOS 100 103-key keyboard.

Key Switch Number

110	112	113	114	115	116	117	118	119	120	121	122	123	75	80	85
1	2	3	4	5	6	7	8	9	10	11	12	13	76	81	86
16	17	18	19	20	21	22	23	24	25	26	42	15	91	96	101
30	31	32	33	34	35	36	37	38	39	40	41	43	92	97	102
44	45	46	47	48	49	50	51	52	53	54	83		93	98	103
58	60	27	28	(*1)	203	61	62	64	55	79	84	89	99	94	104

Figure 69. SurePOS 100 103-key keyboard switches

Table 231 shows the key scan codes for the SureOne and SurePOS 100 keyboards.

Table 231. SureOne/SurePOS 100 keyboard scan codes

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
1	0x29	0xA9	0x29	
2	0x02	0x82	0x02	
3	0x03	0x83	0x03	
4	0x04	0x84	0x04	
5	0x05	0x85	0x05	
6	0x06	0x86	0x06	
7	0x07	0x87	0x07	
8	0x08	0x88	0x08	
9	0x09	0x89	0x09	
10	0x0a	0x8a	0x0a	
11	0x0b	0x8b	0x0b	
12	0x0c	0x8c	0x0c	
13	0x0d	0x8d	0x0d	
15	0x0e	0x8e	0x0e	
16	0x0f	0x8f	0x0f	
17	0x10	0x90	0x10	
18	0x11	0x91	0x11	
19	0x12	0x92	0x12	
20	0x13	0x93	0x13	
21	0x14	0x94	0x14	
22	0x15	0x95	0x15	
23	0x16	0x96	0x16	
24	0x17	0x97	0x17	
25	0x18	0x98	0x18	
26	0x19	0x99	0x19	
27	0x1a	0x9a	0x1a	
28	0x1b	0x9b	0x1b	
29	0x2b	0xab	0x2b	
30	0x3a	0xba	0x3a	
31	0x1e	0x9e	0x1e	
32	0x1f	0x9f	0x1f	
33	0x20	0xa0	0x20	
34	0x21	0xa1	0x21	
35	0x22	0xa2	0x22	
36	0x23	0xa3	0x23	
37	0x24	0xa4	0x24	
38	0x25	0xa5	0x25	

Table 231. SureOne/SurePOS 100 keyboard scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
39	0x26	0xa6	0x26	
40	0x27	0xa7	0x27	
41	0x28	0xa8	0x28	
43	0x1c	0x9c	0x1c	
44	0x2a	0xaa	0x2a	
46	0x2c	0xac	0x2c	
47	0x2d	0xad	0x2d	
48	0x2e	0xae	0x2e	
49	0x2f	0xaf	0x2f	
50	0x30	0xb0	0x30	
51	0x31	0xb1	0x31	
52	0x32	0xb2	0x32	
53	0x33	0xb3	0x33	
54	0x34	0xb4	0x34	
55	0x35	0xb5	0x35	
58	0x1d	0x9d	0x1d	
60	0x38	0xb8	0x38	
61	0x39	0xb9	0x39	
62	0xe0 0x38	0xe0 0xb8	0x01 0x38	
64	0xe0 0x1d	0xe0 0x9d	0x01 0x1d	
75	0xe0 0x52	0xe0 0xd2	0x01 0x52	
76	0xe0 0x53	0xe0 0xd3	0x 010x53	
79	0xe0 0x4b	0xe0 0xcb	0x01 0x4b	
80	0xe0 0x47	0xe0 0xc7	0x01 0x47	
81	0xe0 0x4f	0xe0 0xcf	0x01 0x4f	
83	0xe0 0x48	0xe0 0xc8	0x01 0x48	
84	0xe0 0x50	0xe0 0xd0	0x01 0x50	
85	0xe0 0x49	0xe0 0xc9	0x01 0x49	
86	0xe0 0x51	0xe0 0xd1	0x01 0x51	
89	0xe0 0x4d	0xe0 0xcd	0x01 0x4d	
91	0x47	0xc7	0x47	
92	0x4b	0xcb	0x4b	
93	0x4f	0xcf	0x4f	
94	0x52	0xd2	0x52	Double "00" disabled in BIOS
94	0x52 0x52	0xd2	0x52	SureOne: Double'00' enabled in BIOS

Table 231. SureOne/SurePOS 100 keyboard scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
94	0x0b 0x8b 0x0b 0x8b		0x0b(down) 0x0b(up) 0x0b(down) 0x0b(up)	SurePOS 100: Double'00' enabled in BIOS
96	0x48	0xc8	0x48	
97	0x4c	0xcc	0x4c	
98	0x50	0xd0	0x50	
99	0x52	0xd2	0x52	
101	0x49	0xc9	0x49	
102	0x4d	0xcd	0x4d	
103	0x51	0xd1	0x51	
104	0x53	0xd3	0x53	
110	0x01	0x81	0x01	
112	0x3b	0xbb	0x3b	
113	0x3c	0xbc	0x3c	
114	0x3d	0xbd	0x3d	
115	0x3e	0xbe	0x3e	
116	0x3f	0xbf	0x3f	
117	0x40	0xc0	0x40	
118	0x41	0xc1	0x41	
119	0x42	0xc2	0x42	
120	0x43	0xc3	0x43	
121	0x44	0xc4	0x44	
122	0x57	0xd7	0x57	
123	0x58	0xd8	0x58	
Pos Shift + Ins		0xe0 0xb7 0xe0 0xaa	0x0137	Key release only
Pos Shift + Home	0x46	0xc6	0x46	
Pos Shift + PgUp	0xe1 0x1d 0x45 0xe1 0x9d 0xc5		0x0145	Key press only
Pos Shift + Esc	0xe1 0x1d 0x45 0xe1 0x9d 0xc5		0x0145	Key press only (SurePOS 100 only)
	+ Ins = (136-PrtScr)			
(*1) Pos Shift		0xe0 0xb7 0xe0 0xaa	0x0137	Key release only
	+ Home = (137-ScrLk)			
(*1)	0x46	0xc6	0x46	
	+ PgUp = (138-pause)			

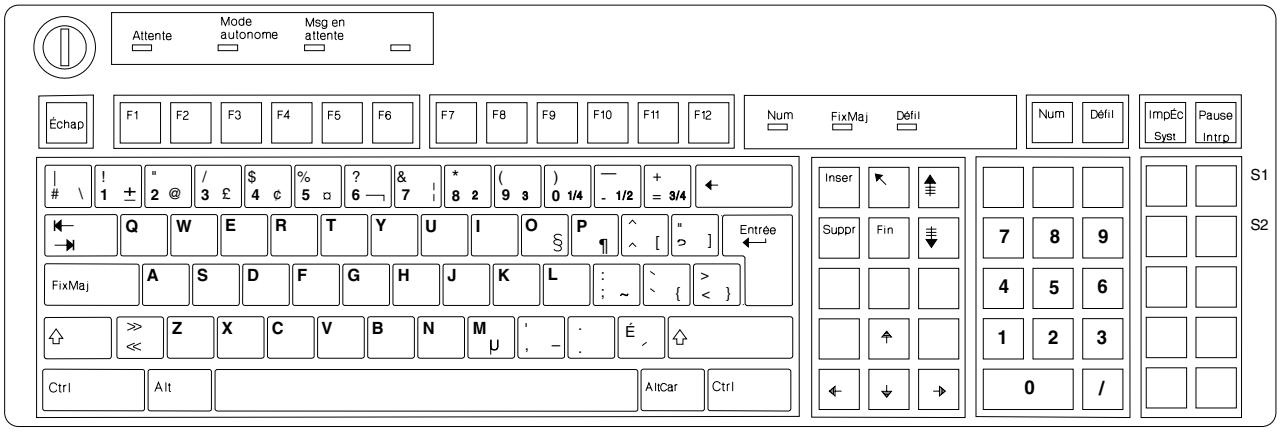
Table 231. SureOne/SurePOS 100 keyboard scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
(*1)	0xe1 0x1d 0x45 0xe1 0x9d 0xc5		0x0145	Key press only

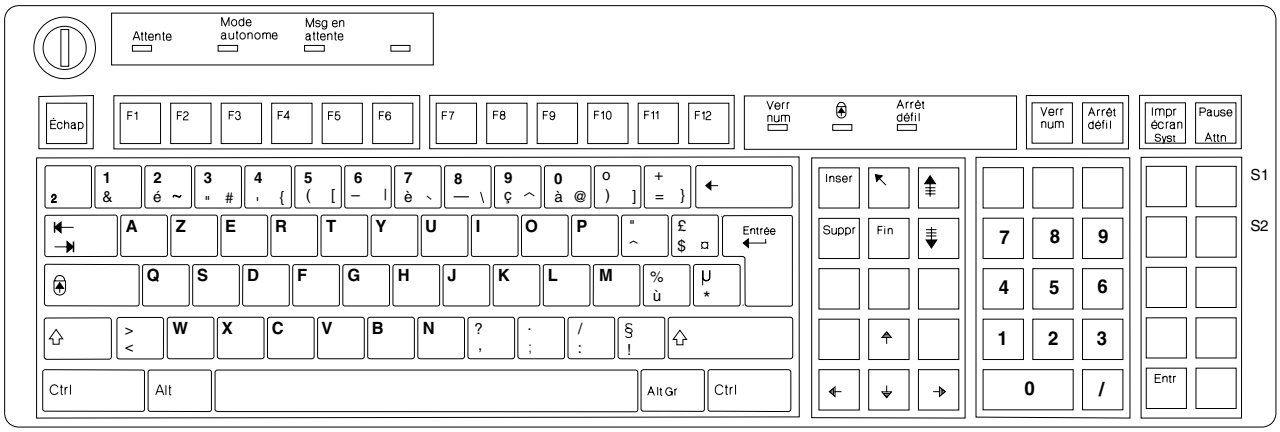
Alphanumeric Point-of-Sale (NANPOS) country-dependent keyboards

The following pages illustrate the keyboards for all the supported NANPOS keyboards.

Canadian French keyboard layout

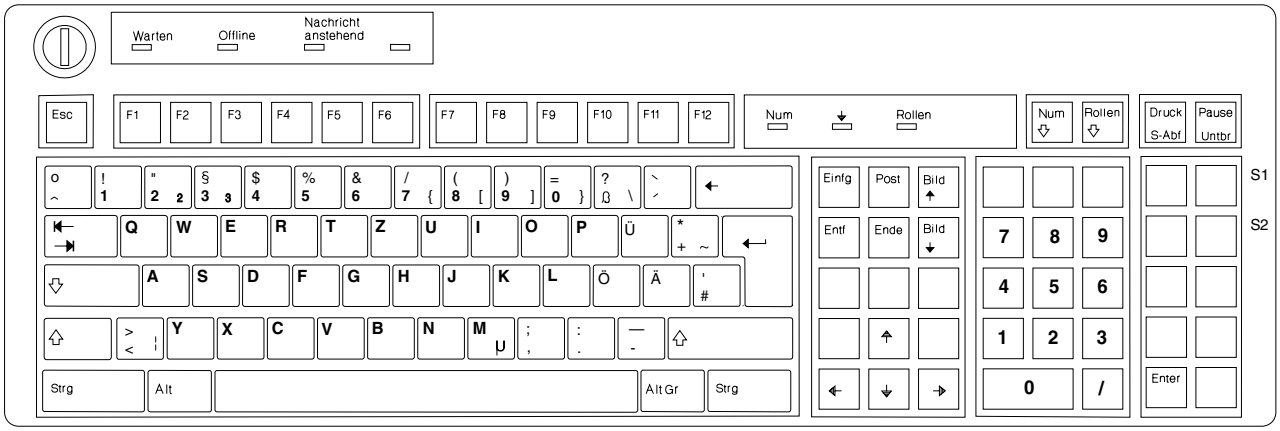


French keyboard layout

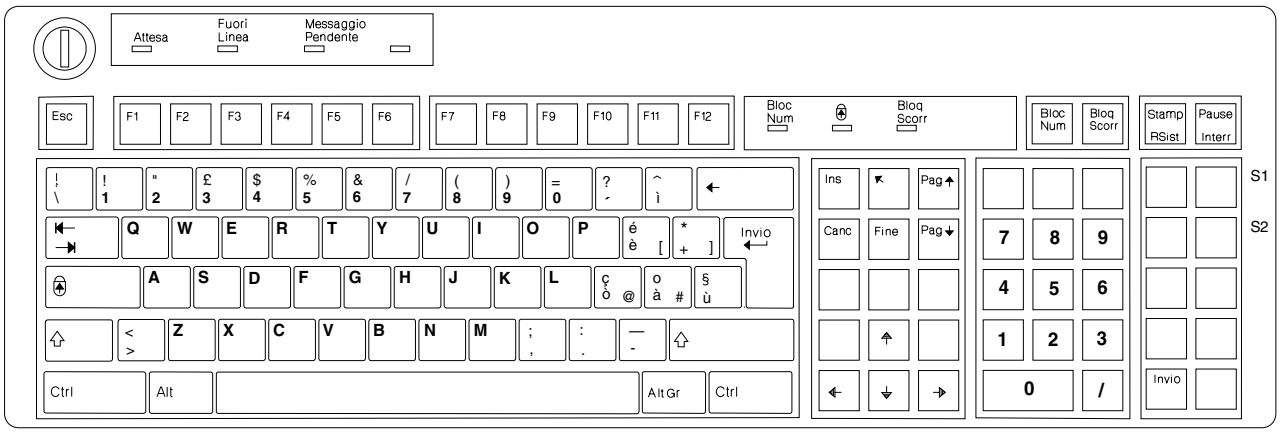


Updated: March 27, 2009

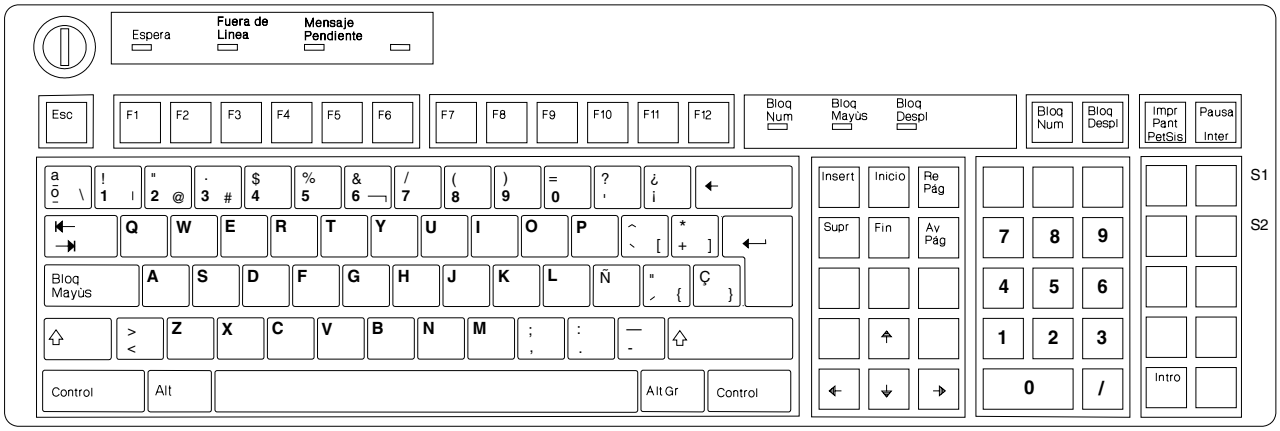
German keyboard layout



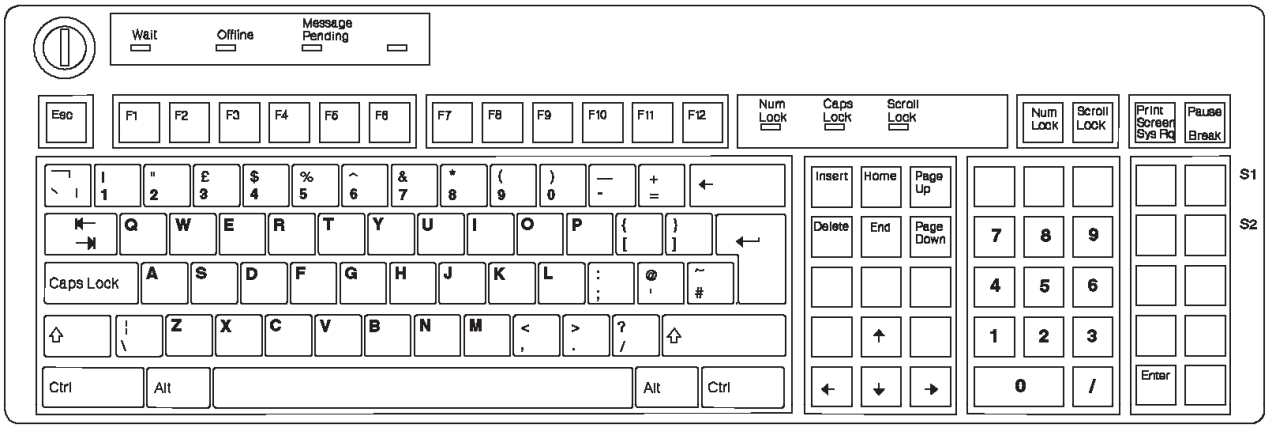
Italian keyboard layout



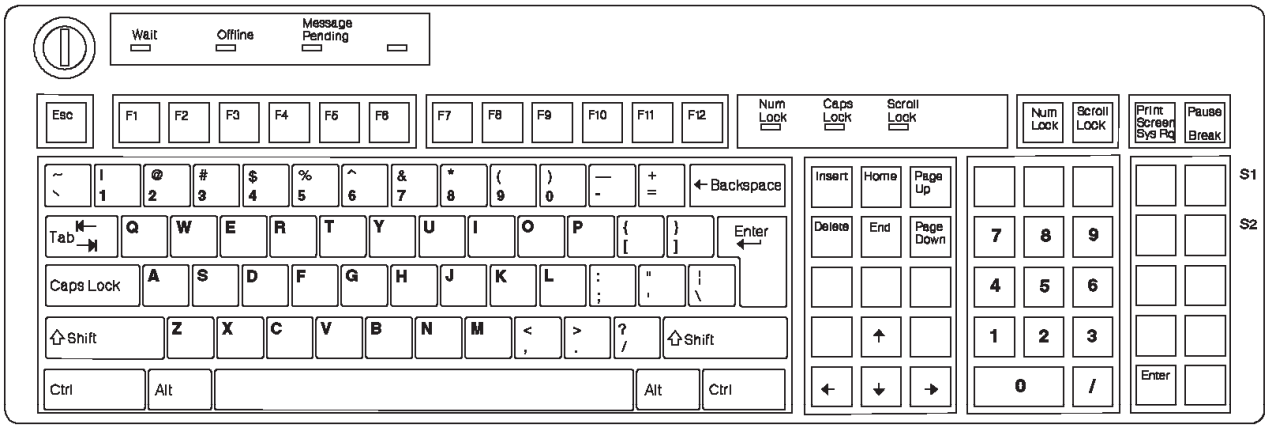
Spanish keyboard layout



U.K. English keyboard layout



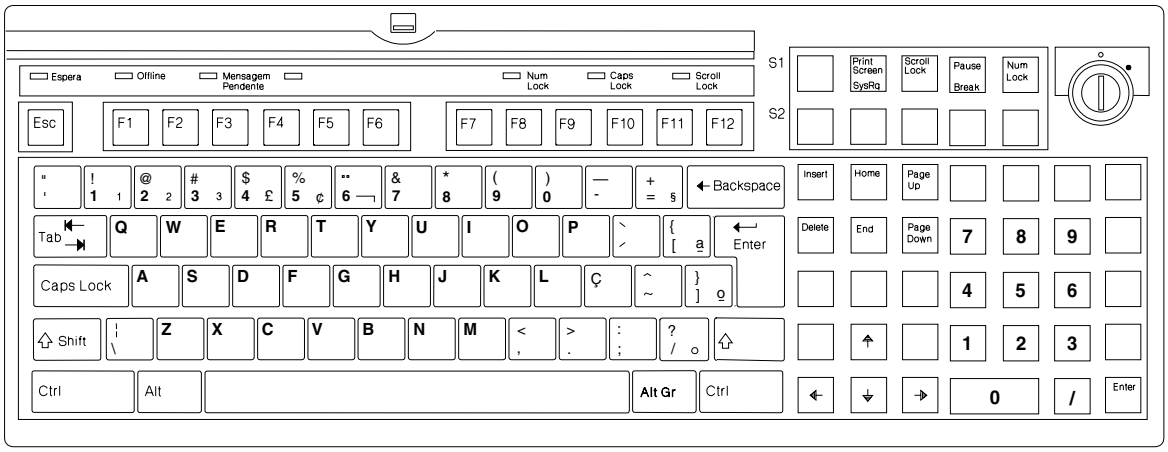
U.S. English keyboard layout



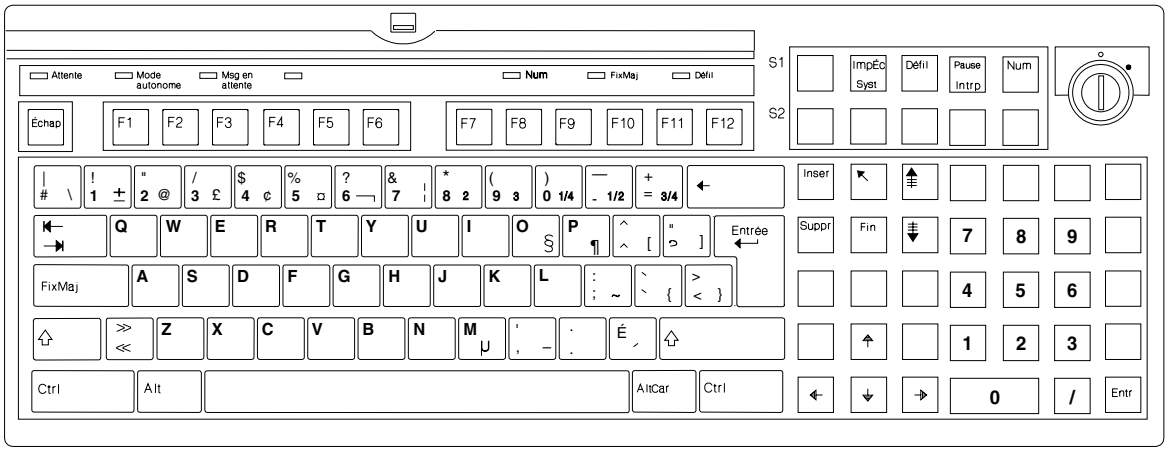
Retail Alphanumeric Point of Sale country dependent keyboards

The following pages illustrate the keyboards for all the supported Retail Alphanumeric Point of Sale Keyboards.

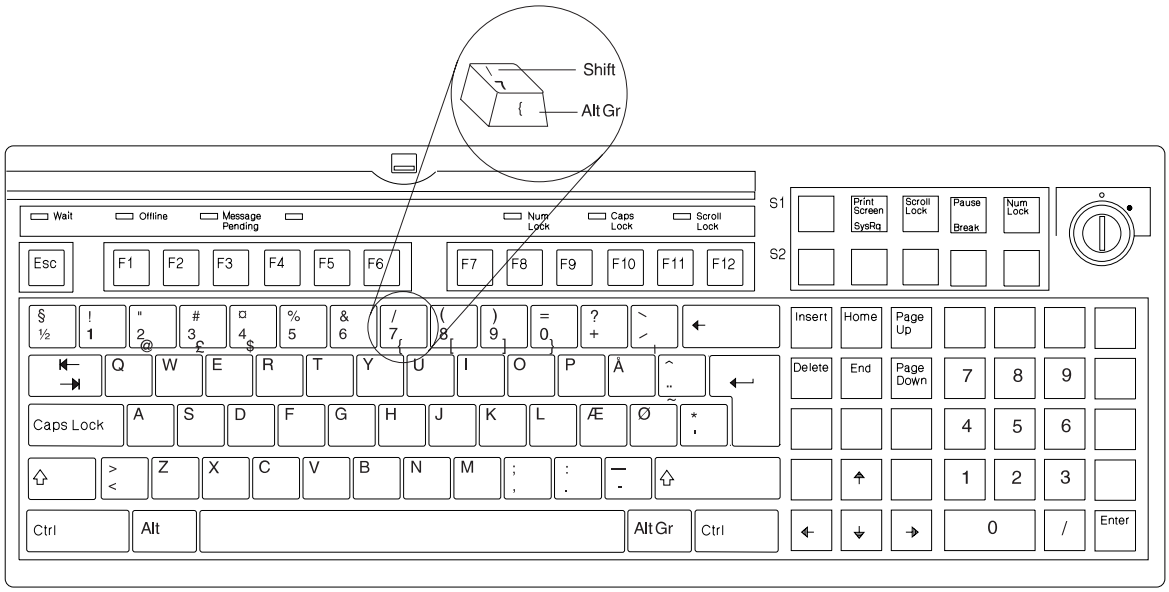
Brazil-Portuguese keyboard layout



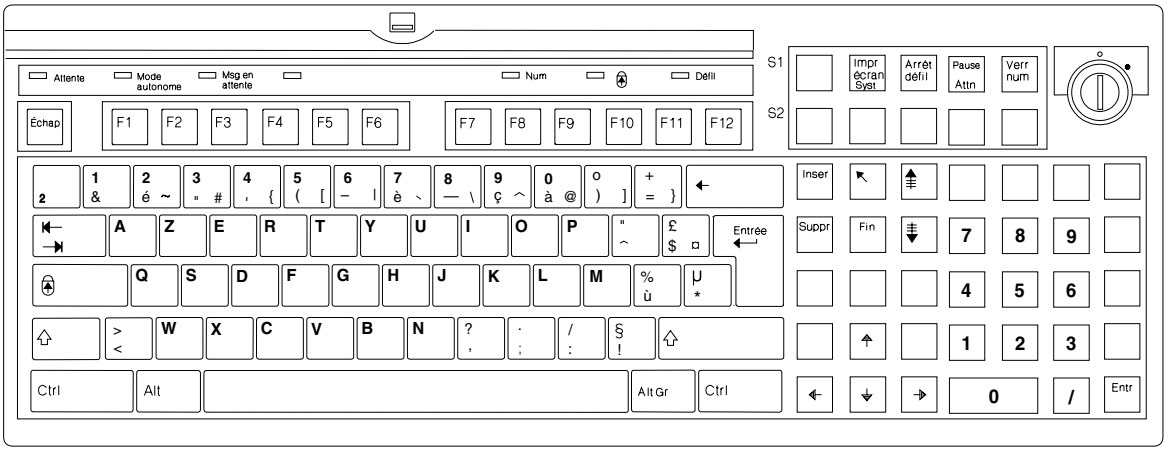
Canadian French keyboard layout



Danish keyboard layout

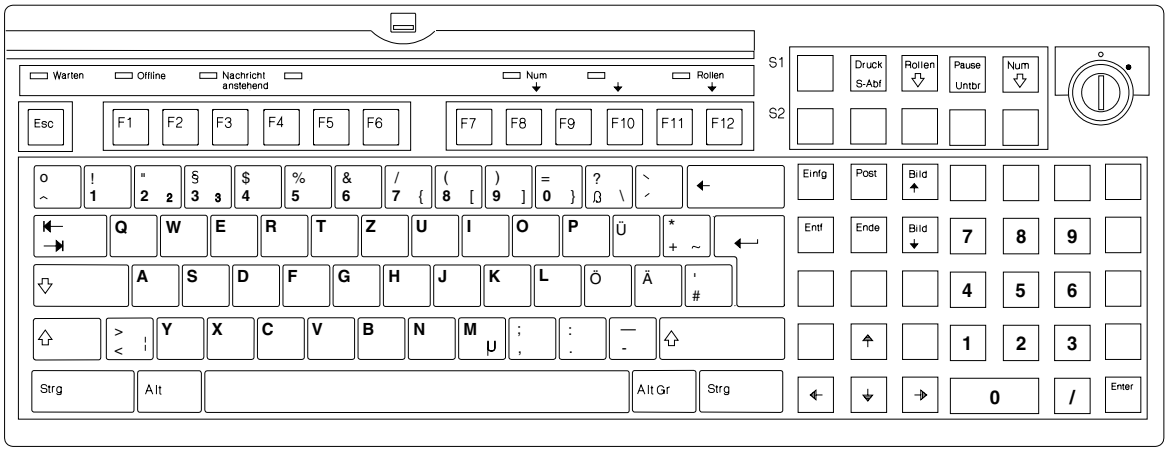


French keyboard layout

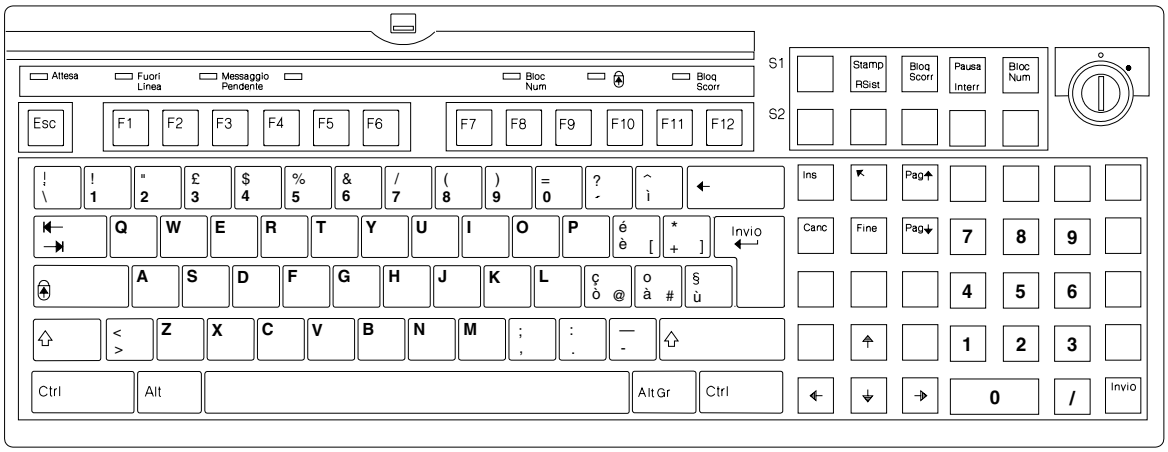


Updated: March 27, 2009

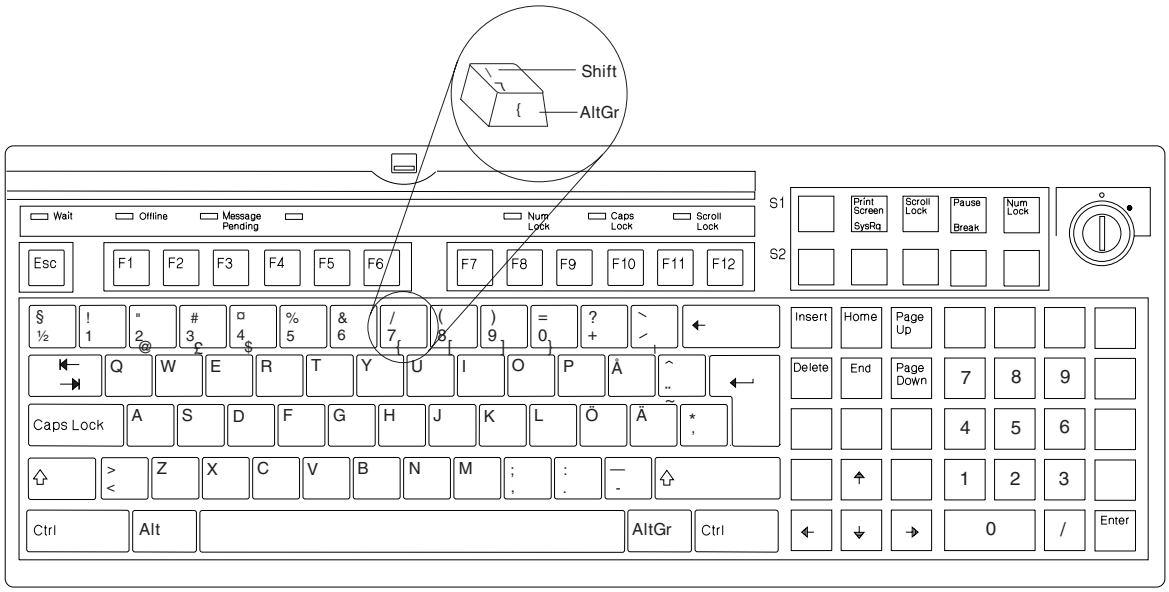
German keyboard layout



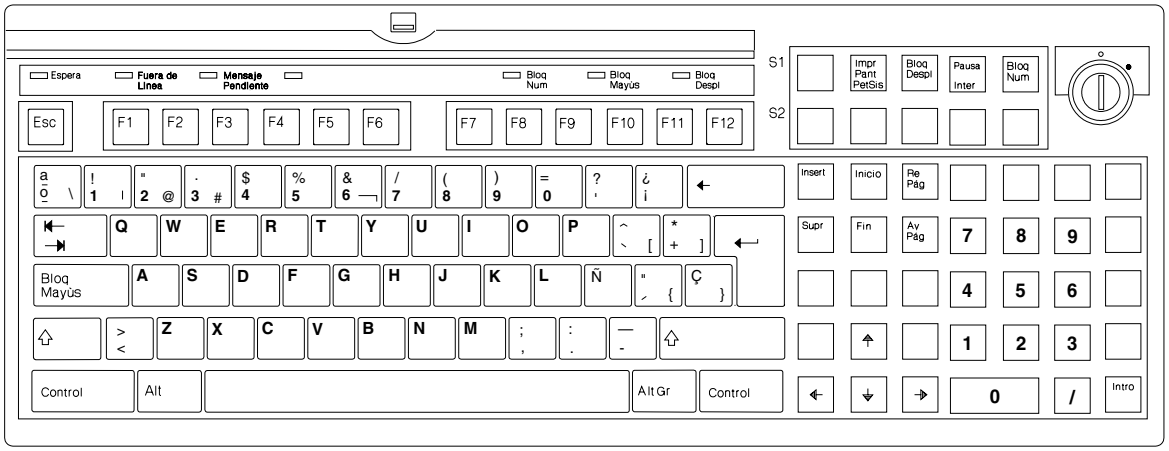
Italian keyboard layout



Norwegian keyboard layout

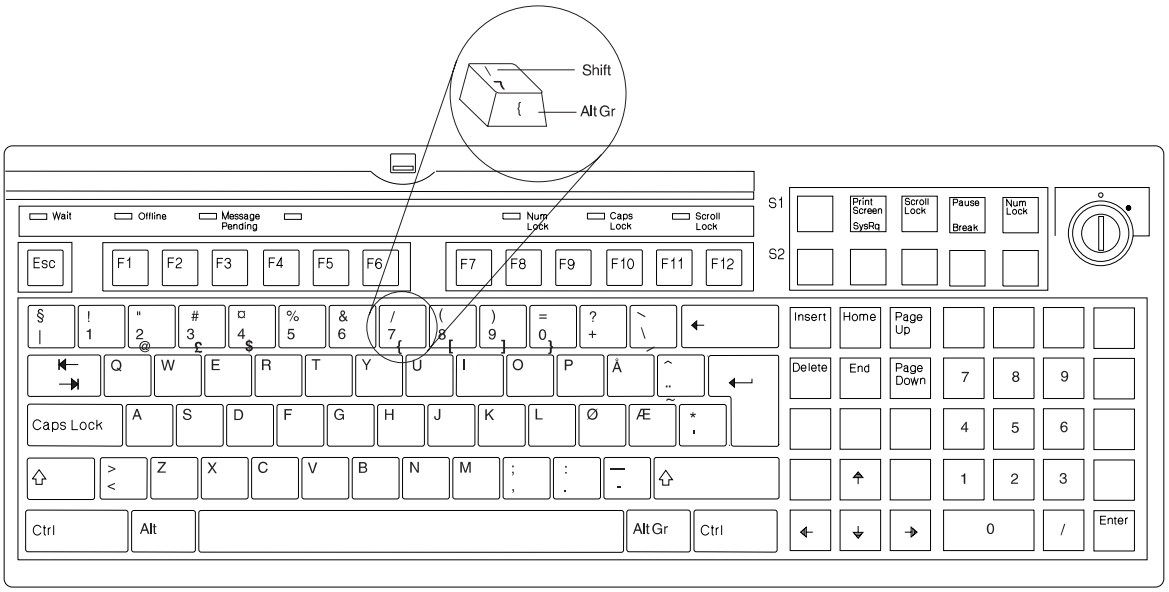


Spanish keyboard layout

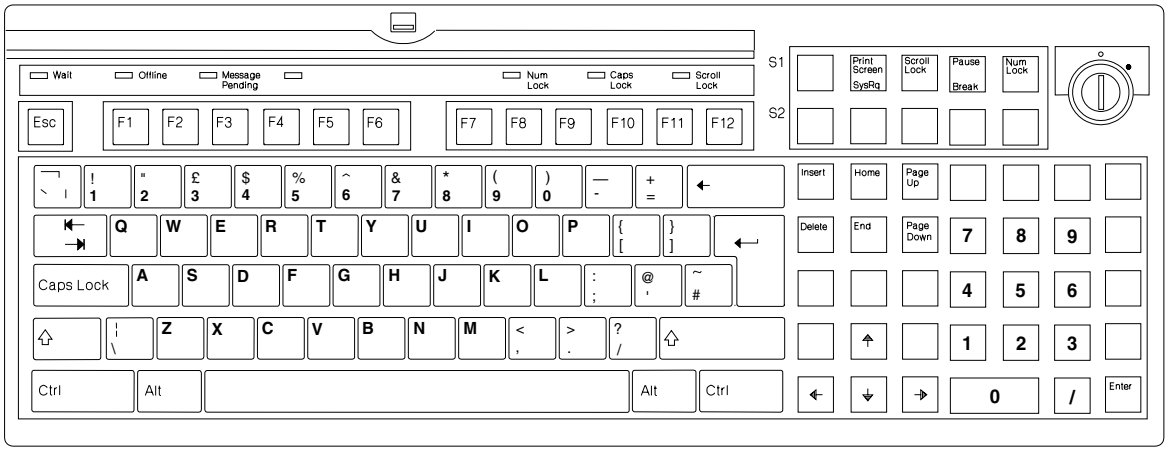


Updated: March 27, 2009

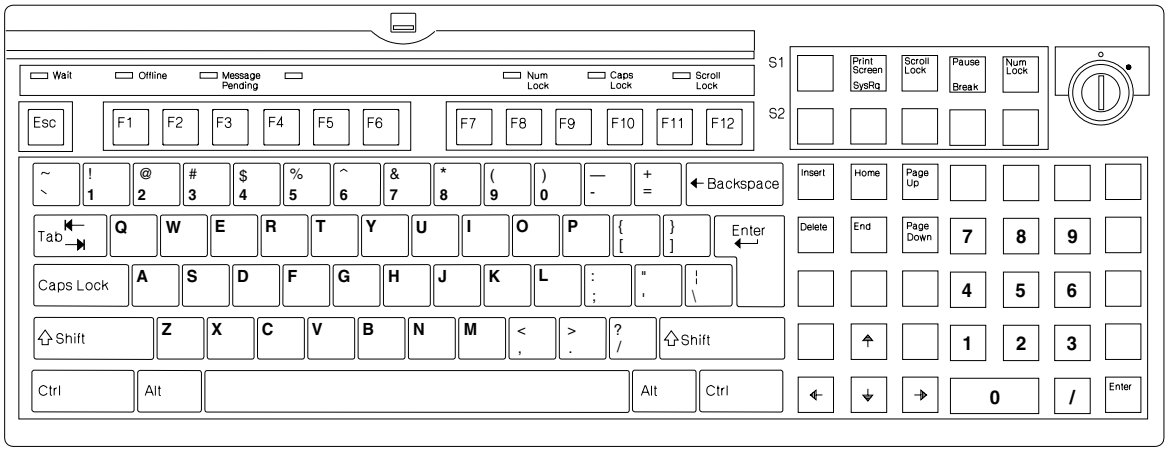
Swedish and Finnish keyboard layout



U.K. English keyboard layout



U.S. English keyboard layout



Compact Alphanumeric Point-of-Sale (CANPOS) keyboard

This section describes the layout and assigned key-switch numbers for the CANPOS keyboards

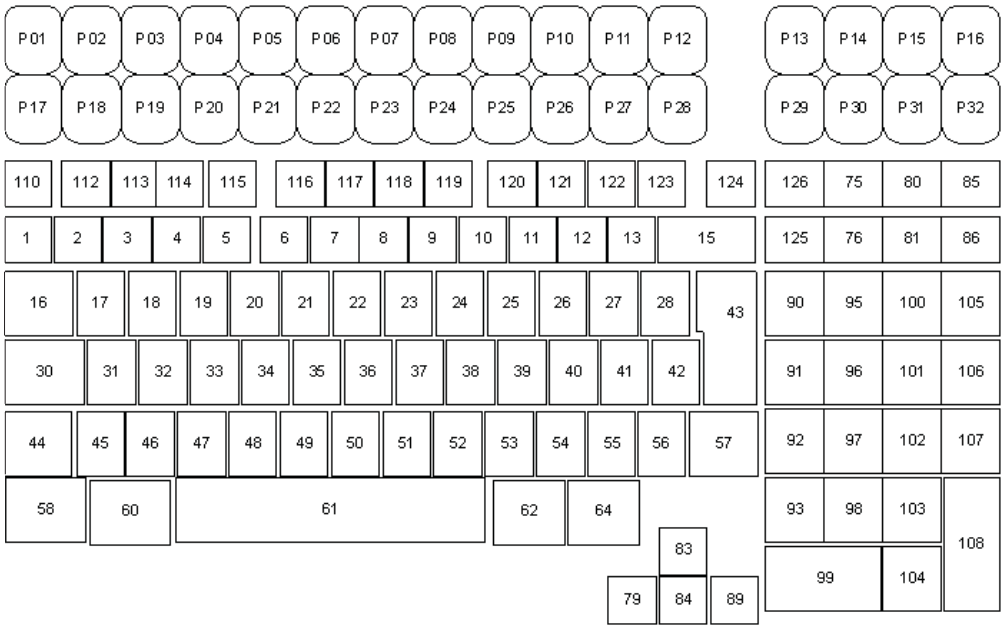
CANPOS keyboards by country

This section describes the keyboard layout for the Compact Alphanumeric Point of Sale (CANPOS) keyboard.

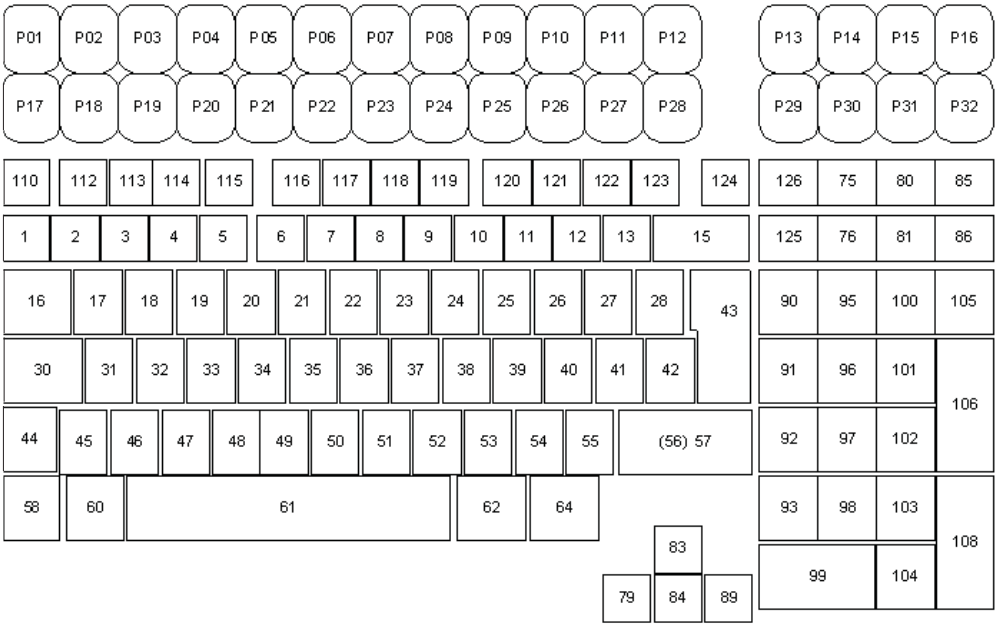
Scan code keyboard assignments

This section shows the CANPOS scan code keyboard assignment by country.

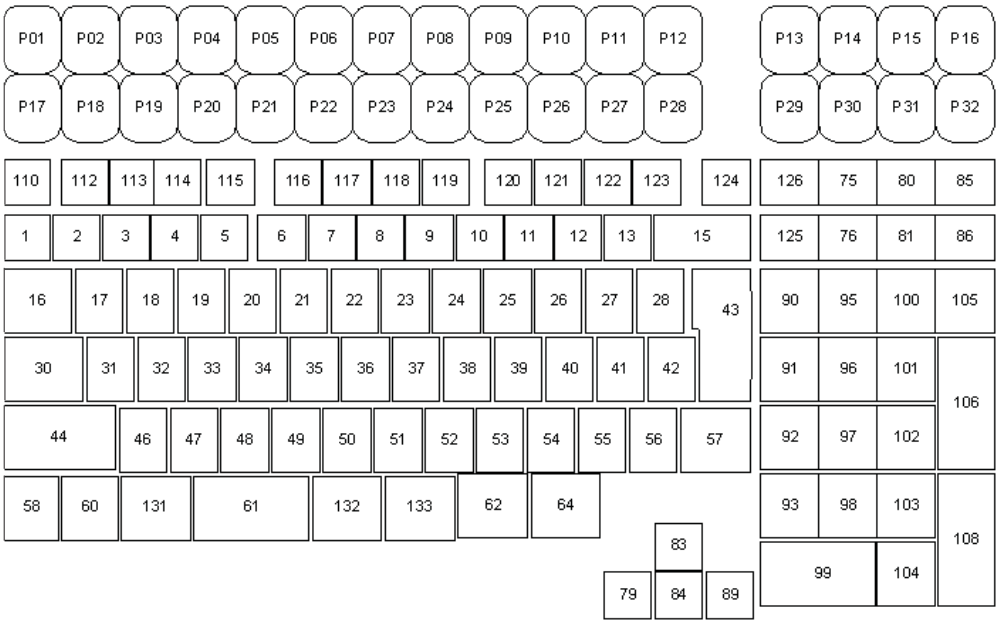
Brazilian keyboard:



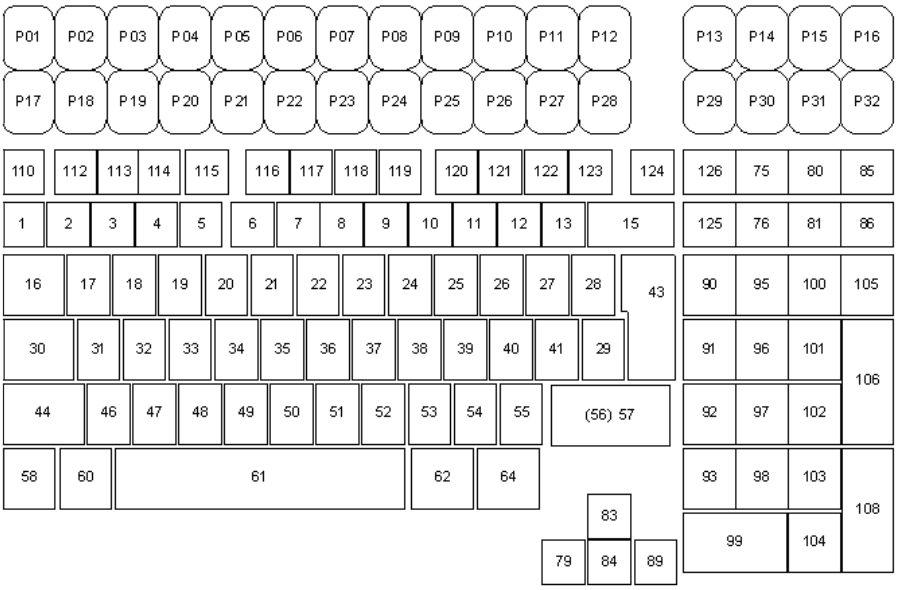
European keyboard:



Japanese keyboard:



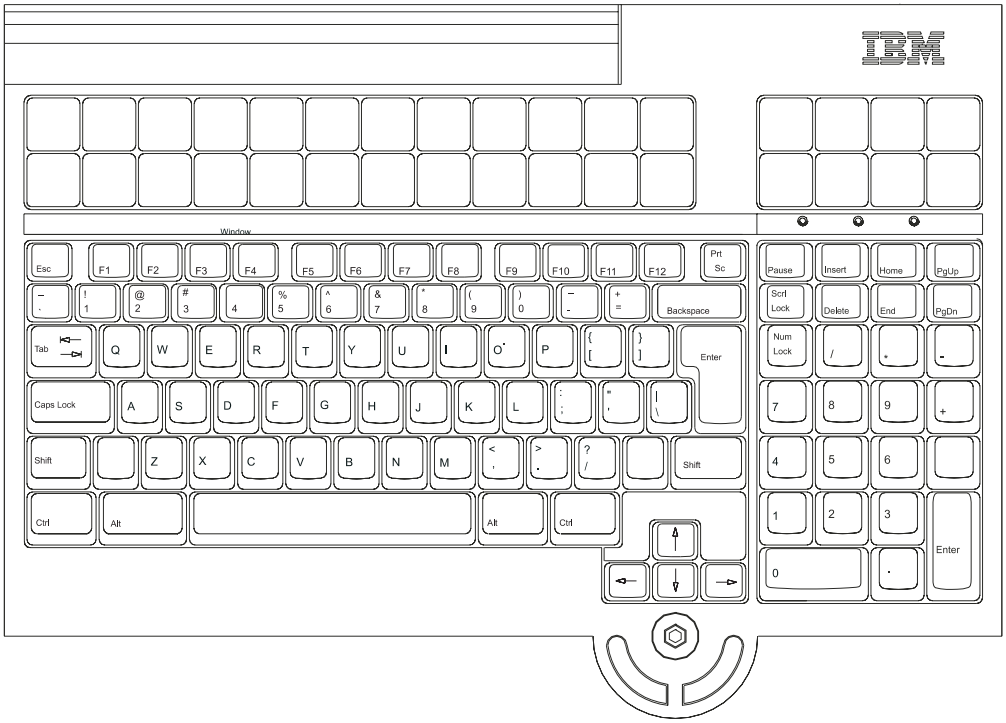
U.S. English keyboard:



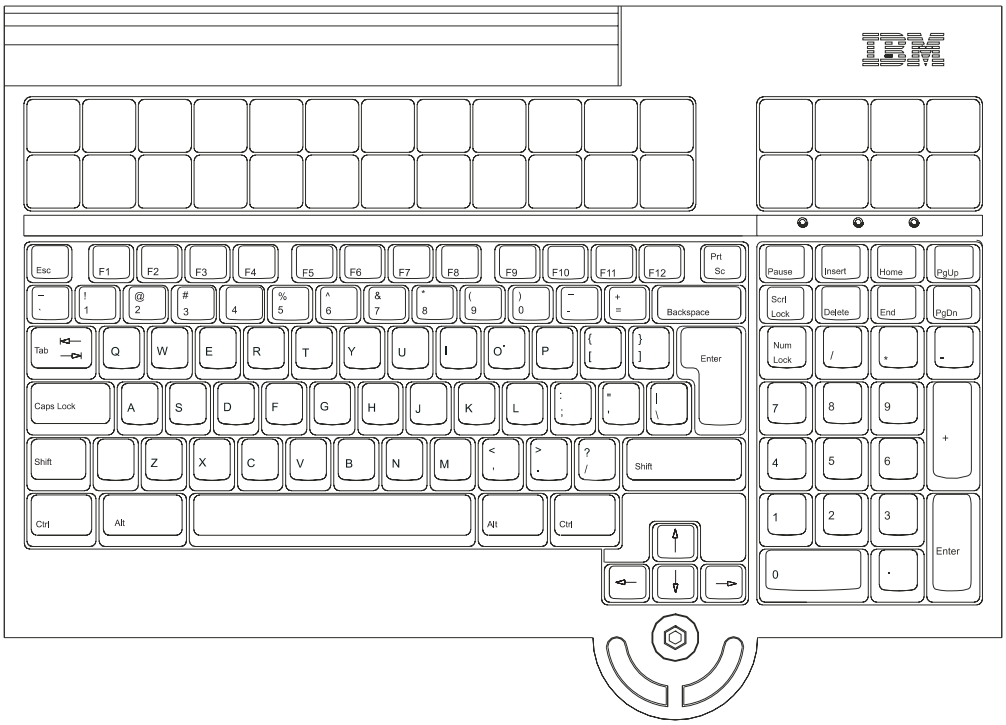
CANPOS country-dependent keyboards

The following pages illustrate the keyboards for all the supported CANPOS keyboards.

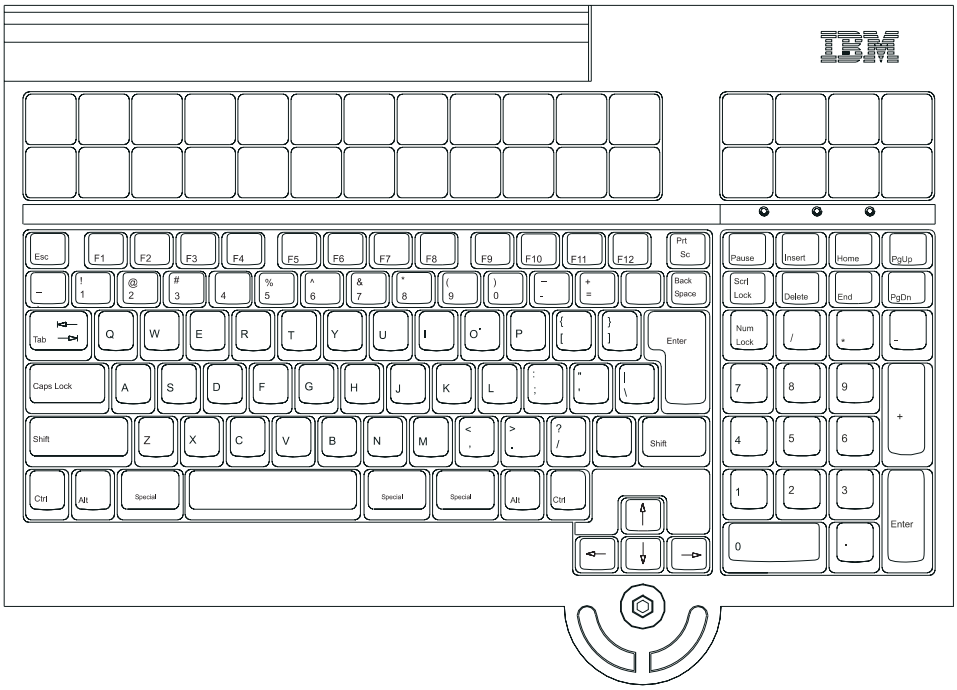
Brazilian keyboard



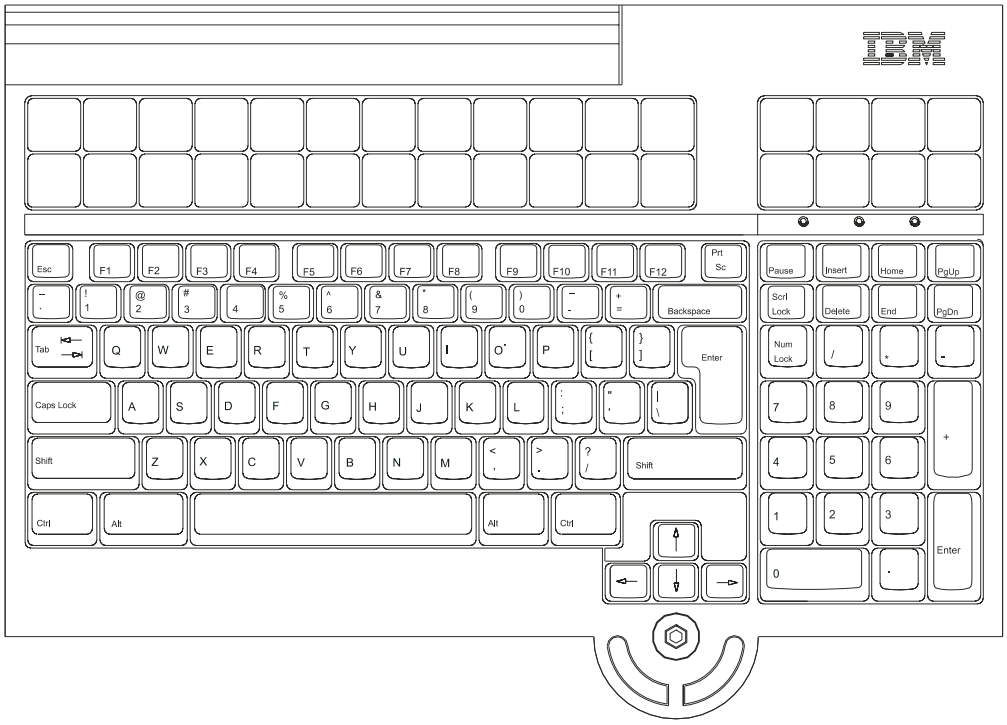
European keyboard



Japanese keyboard



U.S. English keyboard



Chapter 21. Character sets for terminal printers and displays

This chapter contains the character sets and the ASCII codes for each character for the following devices:

- Alphanumeric Display
- Operator Display
- Shopper Display
- Character/Graphics Display
- PLU Keyboard/Display-III
- 40-Character Vacuum Fluorescent Display II
- Two-Sided Vacuum Fluorescent Display II
- 40-Character Liquid Crystal Display
- 2x20 Character VFD Customer Display
- 1x11 LED Display
- IBM Model 3 Printer
- IBM Model 3F Fiscal Printer
- IBM Model 3R Printer
- IBM Model 4 Printer
- IBM Model 4A Printer
- IBM Model 4R Printer
- IBM 4610 SureMark Point of Sale Printer Models TI1, TI2, TI3, TI4, TI5, TI8, TI9, TG3, TG4, TG8, TG9, TF6, TF7, TM6, TM7
- IBM 4610 Fiscal Printer Models K*xx* and G*xx*
- IBM 4689 Point of Sale Printer Models 001, 002, 301, 3G1, 3M1, TD5

The following code pages are supported:

- 101 OEM 7-Segment Alphanumeric font
- 437 U.S. English
- 808 Cyrillic
- 819 ISO-8859-1 / 1252
- 850 OEM Latin-1
- 852 OEM Latin-2
- 855 OEM Cyrillic
- 857 OEM Turkish
- 858 OEM Latin-1 + Euro
- 860 OEM Portuguese
- 861 OEM Icelandic
- 862 OEM Israel
- 863 OEM French Canadian
- 864 OEM Arabic
- 865 OEM Nordic
- 866 OEM Russian
- 869 OEM Green
- 897 Single-byte Japanese Katakana (also used in CP 932)
- 926 Double-byte Korean Hangul
- 932 Japanese Shift-JIS (see 2 on page 338)
- 936 Simplified Chinese
- 949 Korean (see 2 on page 338)
- 950 Traditional Chinese Big5 (see 3 on page 338)
- 951 (Special symbols Alphanumeric/Numerics/Jamo/Hiragana/Hanja)
- 1116 IBM Estonia
- 1117 IBM Latvia
- 1118 IBM Lithuania
- 1119 IBM Russia/Lithuania
- 1250 MS Windows - Central Europe

- 1251 MS Windows - Cyrillic
- 1252 MS Windows - Latin 1 / ISO 8859-1 / 819
- 1253 MS Windows - Greek
- 1254 MS Windows - Turkish / ISO 8859-4
- 1257 MS Windows - Baltic
- 1361 Korean - Hohab
- 1381 Simplified Chinese GB2312 (see 3)

Notes:

1. See “Character sets supported by line display devices for JPOS” on page 104 for the list of line displays and their supported code pages.
2. Code pages 932 and 949 are mixed, single-byte/double-byte code pages.
   - Code page 932 contains the following:
     - Single-byte character code page 897
     - Double-byte character code page 301
   - Code page 949 contains the following:
     - Single-byte character code page 1088
     - Double-byte character code page 951
3. Code pages 950 and 1381 are supported, but are too large to document in this book.

Code Page 101

LED 1X11 LINE DISPLAY CHARACTER SET AND EXPECTED OUTPUT																
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
20H				-												
														_		
30H	0	1	2	3	4	5	6	7	8	9				=		
40H		A	B	C	D	E	F	G	H	I	J		L		N	O
50H	P	Q	R	S	T	U				Y						_
60H		a	b	c	d	e	f	g	h	i	j		l		n	o
70H	p	q	r	s	t	u				y						

Figure 70. Code page 101 chart

Code Page 301

8140	8141	8142	8143	8144	8145	8146	8147	8148	8149	814A	814B	814D	814F	8150	8151
	、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨	―
815B	815E	8162	8169	816A	816D	816E	816F	8170	8175	8176	817B	817C	8181	8183	8184
ー	／	｜	（	）	［	］	｛	｝	「	」	＋	－	＝	＜	＞
818C	818D	818F	8190	8193	8194	8195	8196	8197	824F	8250	8251	8252	8253	8254	8255
'	"	￥	＄	％	＃	＆	＊	＠	０	１	２	３	４	５	６
8256	8257	8258	8260	8261	8262	8263	8264	8265	8266	8267	8268	8269	826A	826B	826C
７	８	９	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ
826D	826E	826F	8270	8271	8272	8273	8274	8275	8276	8277	8278	8279	8281	8282	8283
Ｎ	Ｏ	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ	ａ	ｂ	ｃ
8284	8285	8286	8287	8288	8289	828A	828B	828C	828D	828E	828F	8290	8291	8292	8293
ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ	ｐ	ｑ	ｒ	ｓ
8294	8295	8296	8297	8298	8299	829A	8340	8341	8342	8343	8344	8345	8346	8347	8348
ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ	ｱ	ｱ	ｲ	ｲ	ｳ	ｳ	ｴ	ｴ	ｵ
8349	834A	834B	834C	834D	834E	834F	8350	8351	8352	8353	8354	8355	8356	8357	8358
ｵ	ｶ	ｶﾞ	ｷ	ｷﾞ	ｸ	ｸﾞ	ｹ	ｹﾞ	ｺ	ｺﾞ	ｻ	ｻﾞ	ｼ	ｼﾞ	ｽ
8359	835A	835B	835C	835D	835E	835F	8360	8361	8362	8363	8364	8365	8366	8367	8368
ｽﾞ	ｾ	ｾﾞ	ｿ	ｿﾞ	ﾀ	ﾀﾞ	ﾁ	ﾁﾞ	ｯ	ﾂ	ﾂﾞ	ﾃ	ﾃﾞ	ﾄ	ﾄﾞ
8369	836A	836B	836C	836D	836E	836F	8370	8371	8372	8373	8374	8375	8376	8377	8378
ﾅ	ﾆ	ﾇ	ﾈ	ﾉ	ﾊ	ﾊﾞ	ﾊﾟ	ﾋ	ﾋﾞ	ﾋﾟ	ﾌ	ﾌﾞ	ﾌﾟ	ﾍ	ﾍﾞ
8379	837A	837B	837C	837D	837E	8380	8381	8382	8383	8384	8385	8386	8387	8388	8389
ﾍﾟ	ﾎ	ﾎﾞ	ﾎﾟ	ﾏ	ﾐ	ﾑ	ﾒ	ﾓ	ｬ	ﾔ	ｭ	ﾕ	ｮ	ﾖ	ﾗ
838A	838B	838C	838D	838F	8392	8393									
ﾘ	ﾙ	ﾚ	ﾛ	ﾜ	ｦ	ﾝ									

Figure 71. Alphanumeric Katakana

829F	82A0	82A1	82A2	82A3	82A4	82A5	82A6	82A7	82A8	82A9	82AA	82AB	82AC	82AD	82AE
ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く	ぐ
82AF	82B0	82B1	82B2	82B3	82B4	82B5	82B6	82B7	82B8	82B9	82BA	82BB	82BC	82BD	82BE
け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ
82BF	82C0	82C1	82C2	82C3	82C4	82C5	82C6	82C7	82C8	82C9	82CA	82CB	82CC	82CD	82CE
ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば
82CF	82D0	82D1	82D2	82D3	82D4	82D5	82D6	82D7	82D8	82D9	82DA	82DB	82DC	82DD	82DE
ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み	む
82DF	82E0	82E1	82E2	82E3	82E4	82E5	82E6	82E7	82E8	82E9	82EA	82EB	82EC	82ED	82EE
め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ	ゐ
82EF	82F0	82F1	88B3	88B5	88B6	88C0	88C3	88C4	88C8	88CB	88CD	88D5	88D9	88DA	88DF
ゑ	を	ん	圧	扱	宛	安	暗	案	以	依	囲	易	異	移	衣
88E1	88E3	88EA	88EB	88F3	88F5	88F6	88F8	88F9	8945	894A	895E	895F	8963	8966	8968
違	医	一	壱	印	員	因	引	飲	右	雨	運	雲	営	映	栄
8969	8974	8976	897A	897E	899B	899D	899E	899F	89A1	89A4	89B3	89B5	89B9	89BA	89BB
永	液	益	越	円	央	往	応	押	横	王	乙	卸	音	下	化
89BC	89BD	89BF	89C1	89C2	89C4	89C6	89C8	89CA	89CD	89CE	89D4	89D7	89D9	89DB	89DF
仮	何	価	加	可	夏	家	科	果	河	火	花	荷	蒲	課	過
89E6	89EE	89EF	89F0	89F1	89F4	89FC	8A43	8A45	8A4A	8A4B	8A4F	8A51	8A65	8A67	8A69
画	介	会	解	回	廻	改	海	界	開	階	外	害	各	拡	格
8A6D	8A70	8A72	8A76	8A7A	8A7C	8A84	8A87	8A94	8AA3	8AAE	8AB7	8ABF	8AC7	8AC8	8AD4
確	角	較	革	額	掛	割	括	株	乾	完	換	漢	管	簡	間
8AD6	8ADF	8AE9	8AED	8AEE	8AFA	8AFC	8B40	8B41	8B43	8B47	8B4C	8B4D	8B5A	8B71	8B78
関	玩	企	器	基	期	寄	機	帰	気	季	記	貴	技	客	休
8B81	8B8B	8B8C	8B8D	8B8E	8B8F	8B90	8B96	8B9B	8B9E	8B9F	8BA4	8BA6	8BAD	8BC6	8BC7
求	給	旧	牛	去	居	巨	許	魚	京	供	共	協	強	業	局
8BCE	8BCF	8BD6	8BD8	8BE0	8BE2	8BE3	8BE6	8BEF	8BF3	8C4A	8C50	8C51	8C53	8C57	8C58
勤	均	禁	筋	金	銀	九	区	具	空	繰	訓	群	郡	係	傾
8C5F	8C60	8C67	8C69	8C6F	8C70	8C76	8C79	8C85	8C87	8C88	8C8B	8C8E	8C8F	8C93	8C94
契	形	携	票	経	継	計	軽	桁	欠	決	結	月	件	兼	券
8C9F	8CA0	8CA2	8CA3	8CA7	8CAF	8CB3	8CB4	8CB8	8CBB	8CBE	8CC0	8CC2	8CC3	8CC4	8CC5
検	権	犬	献	県	険	元	原	減	現	言	限	個	古	呼	固

Figure 72. Kanji (1 of 3)

8CC9	8CDA	8CDC	8CDF	8CE0	8CE3	8CE4	8CEB	8CF0	8CF5	8CF6	8CF8	8CFA	8CFB	8CFC	8D46
庫	顧	五	午	呉	後	御	誤	交	光	公	効	厚	口	向	考
8D48	8D4C	8D54	8D58	8D62	8D72	8D73	8D76	8D80	8D82	8D86	8D87	8D8F	8D90	8D95	8D9E
工	広	控	更	甲	荒	行	貢	項	高	号	合	刻	告	黒	込
8DA1	8DB6	8DB7	8DB8	8DC0	8DC2	8DC3	8DC4	8DC5	8DCB	8DCC	8DCE	8DCF	8DD5	8DD7	8DD8
今	左	差	査	座	債	催	再	最	才	採	歳	済	祭	細	菜
8DDC	8DDD	8DDE	8DE3	8DEC	8DED	8DF0	8E44	8E47	8E4F	8E51	8E52	8E59	8E5A	8E63	8E64
剤	在	材	阪	作	削	昨	札	雑	三	参	山	産	算	残	仕
8E67	8E6C	8E6D	8E6E	8E71	8E73	8E75	8E76	8E77	8E78	8E7D	8E7E	8E80	8E81	8E84	8E85
使	四	士	始	子	市	志	思	指	支	枝	止	死	氏	私	糸
8E86	8E8E	8E8F	8E91	8E96	8E97	8E9A	8E9D	8E9E	8E9F	8EA1	8EA6	8EA8	8EA9	8EAE	8EB5
紙	試	誌	資	事	似	字	持	時	次	治	示	耳	自	式	七
8EB8	8EBA	8EBF	8EC0	8ECA	8ECC	8ED0	8ED2	8ED4	8ED8	8EE6	8EE8	8EED	8EF0	8EF1	8EF3
失	室	質	実	写	捨	社	者	車	借	取	手	種	酒	首	受
8EFB	8EFC	8F48	8F49	8F4B	8F54	8F57	8F5A	8F5B	8F5C	8F5D	8F63	8F64	8F68	8F6A	8F6F
収	周	秋	終	習	週	集	住	充	十	従	縦	重	宿	祝	出
8F74	8F7B	8F80	8F83	8F87	8F88	8F89	8F8A	8F91	8F94	8F97	8F9C	8FA4	8FAC	8FAD	8FB3
春	旬	準	純	順	処	初	所	書	諸	女	除	商	小	少	承
8FB5	8FBA	8FC1	8FC6	8FCC	8FCF	8FD8	8FE1	8FE3	8FEA	8FED	8FEE	8FFC	9046	9048	904D
招	昭	消	照	称	粧	証	障	上	場	常	情	飾	色	食	信
9051	9055	9056	9058	905B	905C	905E	905F	9060	9061	9065	9066	9069	906A	906C	907D
寝	振	新	森	深	申	真	神	素	紳	親	診	進	針	人	図
9085	9094	90A7	90AC	90AE	90B0	90B3	90B4	90B6	90B7	90B8	90BB	90BF	90C2	90C5	90CD
水	数	制	成	整	晴	正	清	生	盛	精	製	請	青	税	析
90CE	90CF	90D0	90D1	90D3	90D4	90D8	90DD	90DF	90E0	90E1	90E6	90E7	90E8	90E9	90EA
石	積	籍	績	責	赤	切	設	節	説	雪	先	千	占	宣	専
90EC	90F4	9149	914B	914E	914F	9153	9162	9165	916E	9177	917D	9180	918A	918D	9195
川	洗	選	銭	鮮	前	全	礎	粗	創	層	挿	操	相	総	装
9196	9197	919D	91A1	91A3	91A6	91AB	91AE	91B1	91B9	91BA	91BC	91BD	91BE	91C5	91CC
走	送	増	贈	促	即	足	属	続	損	村	他	多	太	打	体
91CE	91D6	91DD	91DE	91E3	91E4	91E5	91EE	91F0	9242	9249	9250	9253	926A	926B	926C
対	替	貸	退	代	台	大	宅	択	達	棚	単	担	男	談	値

Figure 73. Kanji (2 of 3)

926D	926E	9285	9286	928D	929A	92A0	92A3	92AC	92B2	92BC	92C0	92C2	92C7	92CA	92DE
知	地	着	中	注	丁	帳	張	町	調	直	貯	陳	追	通	鈎
92E1	92E2	92E8	92F1	92F9	9350	9358	935D	935F	9360	9363	9364	936E	936F	9373	9378
低	停	定	提	訂	撤	店	転	点	伝	田	電	渡	登	都	度
9379	937E	9380	9387	938C	9394	9396	9399	939A	939E	93A2	93AA	93AE	93AF	93B9	93BA
土	冬	凍	島	東	灯	当	等	答	到	討	頭	動	同	道	銅
93BE	93C1	93C2	93C6	93C7	93CD	93DC	93E0	93EC	93F1	93F3	93F7	93FA	93FB	93FC	9443
得	特	督	独	読	届	畳	内	南	二	弐	肉	日	乳	入	任
9446	944E	945B	945C	9467	946E	9470	947A	947B	9483	9484	9491	9492	94A0	94AA	94AD
認	年	納	能	波	馬	廃	配	倍	買	売	泊	白	箱	八	発
94BB	94BC	94BD	94CA	94CC	94D4	94E4	94EF	94F1	94F5	9553	9557	9558	9559	955B	955C
判	半	反	般	販	番	比	費	非	備	百	標	氷	漂	票	表
9562	9569	9572	9573	9574	9576	9577	957B	9583	958A	9594	959E	95A5	95A8	95AA	95B6
秒	品	瓶	不	付	夫	婦	府	父	賦	部	服	払	物	分	文
95B8	95B9	95BD	95C2	95C4	95CA	95CF	95D0	95D4	95D6	95D9	95DB	95DC	95E2	95E5	95E9
丙	併	平	閉	米	別	変	片	返	便	弁	保	舗	補	墓	慕
95EA	95EF	95F1	95F2	95FB	9640	966B	967B	9687	9688	9694	9696	969C	96A1	96A2	96B3
母	包	報	奉	方	法	北	本	枚	毎	又	末	万	味	未	無
96BC	96BD	96BE	96C6	96CA	96CD	96D1	96D8	96DA	96DF	96E2	96E5	96EC	96EE	96F1	96F2
名	命	明	免	面	模	毛	木	目	戻	問	門	野	矢	約	薬
96F3	96FB	9746	974C	9752	9758	975B	975C	975D	975E	9761	9765	976A	976C	976D	9770
訳	油	友	有	由	郵	夕	予	余	与	預	容	曜	様	洋	用
9776	9782	978A	978E	9790	9797	9798	979D	97A0	97A6	97A7	97AA	97B9	97BC	97BF	97CA
要	翌	頼	落	乱	覧	利	理	裏	卒	立	略	了	両	料	量
97CC	97CD	97DD	97DE	97E1	97E2	97F1	97FB	9841	985A	985E	9861	9862			
領	力	累	類	例	冷	列	練	連	六	録	和	話			

Figure 74. Kanji (3 of 3)

Code Page 437

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	╨ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	╤ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	ª SM210000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ÿ LY170000	¿ SP160000	╕ SF220000	╚ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	⌐ SM680000	╣ SF230000	╔ SF390000	╘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	║ SF240000	╩ SF400000	╒ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	¢ SC040000	½ NF010000	╗ SF250000	╦ SF410000	╘ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▬ SF570000	∞ SA450000	n LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	¥ SC050000	¡ SP030000	╜ SF270000	═ SF430000	╛ SF580000	ϕ GF010001	2 ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	₧ SC060000	« SP170000	╞ SF280000	╬ SF440000	╘ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	» SP180000	╓ SF030000	╧ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00437

Code Page 850

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	ð LD630000	Ó LO120000	(SHY) SP320000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	Ð LD620000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	Ê LE160000	Ô LO160000	═ SM100000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	Ë LE180000	Ò LO140000	¾ NF050000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	È LE140000	õ LO190000	¶ SM250000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	Á LA120000	┼ SF050000	ı LI610000	Õ LO200000	§ SM240000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	ª SM210000	Â LA160000	ã LA190000	Í LI120000	µ SM170000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	À LA140000	Ã LA200000	Î LI160000	þ LT630000	¸ SD410000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ÿ LY170000	¿ SP160000	© SM520000	╚ SF380000	Ï LI180000	Þ LT640000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	® SM530000	╣ SF230000	╗ SF390000	┘ SF040000	Ú LU120000	¨ SD170000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	╢ SF240000	╬ SF400000	┌ SF010000	Û LU160000	• SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	ø LO610000	½ NF010000	╖ SF250000	╦ SF410000	█ SF610000	Ù LU140000	¹ ND011000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╜ SF260000	╠ SF420000	▬ SF570000	ý LY110000	³ ND031000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	Ø LO620000	¡ SP030000	¢ SC040000	═ SF430000	¦ SM650000	Ý LY120000	² ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	× SA070000	« SP170000	¥ SC050000	╝ SF440000	Ì LI140000	¯ SM150000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	» SP180000	╕ SF030000	¤ SC010000	▬ SF600000	´ SD110000	(RSP) SP300000

Code Page 00850

Code Page 852

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	đ LD610000	Ó LO120000	(SH̄Y) SP320000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	Ĺ LL120000	í LI110000	▒ SF150000	┴ SF070000	Đ LD620000	ß LS610000	″ SD250000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Í LL110000	ó LO110000	▓ SF160000	┬ SF060000	Ď LD220000	Ô LO160000	¸ SD430000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	Ë LE180000	Ń LN120000	ˇ SD210000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	Ą LA440000	┤ SF090000	─ SF100000	ď LD210000	ń LN110000	˘ SD230000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	ů LU270000	Ľ LL220000	ą LA430000	Á LA120000	┼ SF050000	Ň LN220000	ň LN210000	§ SM240000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	ć LC110000	ľ LL210000	Ž LZ220000	Â LA160000	Ă LA240000	Í LI120000	Š LS220000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	Ś LS120000	ž LZ210000	Ě LE220000	ă LA230000	Î LI160000	š LS210000	¸ SD410000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ł LL610000	ś LS110000	Ę LE440000	Ş LS420000	╚ SF380000	ě LE210000	Ŕ LR120000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	ę LE430000	╣ SF230000	╗ SF390000	┘ SF040000	Ú LU120000	¨ SD170000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	Ő LO260000	Ü LU180000		║ SF240000	╝ SF400000	┐ SF010000	ŕ LR110000	• SD290000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ő LO250000	Ť LT220000	ź LZ110000	╖ SF250000	╕ SF410000	■ SF610000	Ű LU260000	ű LU250000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	ť LT210000	Č LC220000	╜ SF260000	╞ SF420000	▬ SF570000	ý LY110000	Ř LR220000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	Ź LZ120000	Ł LL620000	ş LS410000	Ż LZ300000	═ SF430000	Ţ LT420000	Ý LY120000	ř LR210000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	× SA070000	« SP170000	ż LZ290000	╬ SF440000	Ů LU280000	ţ LT410000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Ć LC120000	č LC210000	» SP180000	╓ SF030000	¤ SC010000	▬ SF600000	´ SD110000	(RSP) SP300000

Code Page 00852

Code Page 855

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►	(SP)	0	@	P	`	p	ђ	љ	а	░	└	л	Я	(SHY)
-1	☺	◄	!	1	A	Q	a	q	Ђ	Љ	А	▒	┴	Л	р	ы
-2	☻	↕	"	2	B	R	b	r	ѓ	ањ	б	▓	┬	м	Р	Ы
-3	♥	‼	#	3	C	S	c	s	Ѓ	Њ	Б	│	├	М	с	з
-4	♦	¶	$	4	D	T	d	t	ё	ћ	ц	┤	─	н	С	З
-5	♣	§	%	5	E	U	e	u	Ё	Ћ	Ц	х	┼	Н	т	ш
-6	♠	▬	&	6	F	V	f	v	є	ќ	д	Х	к	о	Т	Ш
-7	•	↨	'	7	G	W	g	w	Є	Ќ	Д	и	К	О	у	э
-8	◘	↑	(	8	H	X	h	x	s	ў	е	И	╚	п	У	Э
-9	○	↓	)	9	I	Y	i	y	S	Ў	Е	╝	╔	┘	ж	щ
-A	◙	→	*	:	J	Z	j	z	i	ц	ф	╜	╩	┌	Ж	Щ
-B	♂	←	+	;	K	[	k	{	І	Ц	Ф	╖	╦	█	в	ч
-C	♀	∟	,	<	L	\	l	|	ї	ю	г	╕	╠	▬	В	Ч
-D	♪	↔	-	=	M	]	m	}	Ї	Ю	Г	й	═	П	ь	§
-E	♫	▲	.	>	N	^	n	~	j	ъ	«	Й	╬	я	Ь	■
-F	☼	▼	/	?	O	_	o	⌂	J	Ъ	»	╗	€	▬	№	(RSP)

Code Page 00855

Code Page 857

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	º SM200000	Ó LO120000	(SHY) SP320000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	ª SM210000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	Ê LE160000	Ô LO160000	
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	Ë LE180000	Ò LO140000	¾ NF050000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	È LE140000	õ LO190000	¶ SM250000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	Á LA120000	┼ SF050000		Õ LO200000	§ SM240000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	Ğ LG240000	Â LA160000	ã LA190000	Í LI120000	µ SM170000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	ğ LG230000	À LA140000	Ã LA200000	Î LI160000		¸ SD410000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	İ LI300000	¿ SP160000	© SM520000	╚ SF380000	Ï LI180000	× SA070000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	® SM530000	╣ SF230000	╗ SF390000	┘ SF040000	Ú LU120000	¨ SD170000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	╠ SF240000	╔ SF400000	┌ SF010000	Û LU160000	• SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	ø LO610000	½ NF010000	╗ SF250000	╦ SF410000	█ SF610000	Ù LU140000	1 ND011000
-C	♀ SM290000	∟ SA420000	, SP080000	< SP030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▬ SF570000	ì LI130000	3 ND031000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ı LI610000	Ø LO620000	¡ SP030000	¢ SC040000	═ SF430000	¦ SM650000	ÿ LY170000	2 ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SP050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	Ş LS420000	« SP170000	¥ SC050000	╬ SF440000	Ì LI140000	¯ SM150000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ş LS410000	» SP180000	╓ SF030000	¤ SC010000	▬ SF600000	´ SD110000	(RSP) SP300000

Code Page 00857

Code Page 858

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	ð LD630000	Ó LO120000	(SHY) SP320000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	Ð LD620000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	Ê LE160000	Ô LO160000	═ SM100000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	Ë LE180000	Ò LO140000	¾ NF050000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	È LE140000	õ LO190000	¶ SM250000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	Á LA120000	┼ SF050000	€ LI610000	Õ LO200000	§ SM240000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	ª SM210000	Â LA160000	ã LA190000	Í LI120000	µ SM170000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	À LA140000	Ã LA200000	Î LI160000	þ LT630000	¸ SD410000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ÿ LY170000	¿ SP160000	© SM520000	╚ SF380000	Ï LI180000	Þ LT640000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	® SM530000	╣ SF230000	╔ SF390000	┘ SF040000	Ú LU120000	¨ SD170000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	╗ SF240000	╩ SF400000	┌ SF010000	Û LU160000	• SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	ø LO610000	½ NF010000	╝ SF250000	╬ SF410000	█ SF610000	Ù LU140000	¹ ND011000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╜ SF260000	╟ SF420000	▬ SF570000	ý LY110000	³ ND031000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	Ø LO620000	¡ SP030000	¢ SC040000	╒ SF430000	¦ SM650000	Ý LY120000	² ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	× SA070000	« SP170000	¥ SC050000	╓ SF440000	Ì LI140000	¯ SM150000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	» SP180000	┐ SF030000	¤ SC010000	▬ SF600000	´ SD110000	(RSP) SP300000

Code Page 00858

Code Page 860

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	╨ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	À LA140000	í LI110000	▒ SF150000	┴ SF070000	╤ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	È LE140000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ã LA190000	õ LO190000	ñ LN190000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	Á LA120000	Ú LU120000	ª SM210000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	Ì LI140000	¿ SP160000	╕ SF220000	╙ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	Ê LE160000	Õ LO200000	Ò LO140000	╣ SF230000	╗ SF390000	┘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	║ SF240000	╚ SF400000	╘ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	Í LI120000	¢ SC040000	½ NF010000	╖ SF250000	╔ SF410000	╘ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	Ô LO160000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▬ SF570000	∞ SA450000	n LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	Ù LU140000	¡ SP030000	╜ SF270000	═ SF430000	╛ SF580000	ϕ GF010001	2 ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ã LA200000	₧ SC060000	« SP170000	╛ SF280000	╬ SF440000	╘ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Â LA160000	Ó LO120000	» SP180000	┐ SF030000	╧ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00860

Code Page 861

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	╨ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	╤ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	Á LA120000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	∫ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	þ LT630000	Í LI120000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌠ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	Ó LO120000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	Ý LY120000	Ú LU120000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ý LY110000	¿ SP160000	╕ SF220000	╚ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	⌐ SM680000	╣ SF230000	╔ SF390000	╘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	╗ SF240000	╩ SF400000	╒ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	Ð LD620000	ø LO610000	½ NF010000	╝ SF250000	╦ SF410000	■ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	ð LD630000	£ SC020000	¼ NF040000	╜ SF260000	╠ SF420000	▬ SF570000	∞ SA450000	ⁿ LN011000
-D	♩ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	Þ LT640000	Ø LO620000	¡ SP030000	╛ SF270000	═ SF430000	▌ SF580000	ϕ GF010001	² ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	Pts SC060000	« SP170000	╞ SF280000	╬ SF440000	▐ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	» SP180000	┐ SF030000	╧ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00861

Code Page 862

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	א HX330000	נ HN010000	á LA110000	░ SF140000	└ SF020000	╨ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ב HB010000	ס HS010000	í LI110000	▒ SF150000	┴ SF070000	╤ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	ג HG010000	ע HX350000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	ד HD010000	ף HP610000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ה HH010000	פ HP010000	ñ LN190000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	ו HW010000	ץ HS610000	Ñ LN200000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	ז HZ010000	צ HS450000	ª SM210000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ח HH450000	ק HQ010000	º SM200000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ט HT450000	ר HR010000	¿ SP160000	╕ SF220000	╙ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	י HY010000	ש HS210000	⌐ SM680000	╣ SF230000	╗ SF390000	┘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	ך HK610000	ת HT010000	¬ SM660000	║ SF240000	╝ SF400000	╜ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	כ HK010000	¢ SC040000	½ NF010000	╗ SF250000	╘ SF410000	╛ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	ל HL010000	£ SC020000	¼ NF040000	╝ SF260000	╞ SF420000	▬ SF570000	∞ SA450000	n LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ם HM610000	¥ SC050000	¡ SP030000	╜ SF270000	╡ SF430000	╛ SF580000	ϕ GF010001	2 ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	מ HM010000	₧ SC060000	« SP170000	╛ SF280000	╬ SF440000	╘ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	ן HN610000	ƒ SC070000	» SP180000	┐ SF030000	╧ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00862

Code Page 863

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	¦ SM650000	░ SF140000	└ SF020000	╩ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	È LE140000	´ SD110000	▒ SF150000	┴ SF070000	╧ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Ê LE160000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	Â LA160000	Ë LE180000	¨ SD170000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	Ï LI180000	¸ SD410000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	¶ SM250000	û LU150000	³ ND031000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	¯ SM150000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	¤ SC010000	Î LI160000	╕ SF220000	╙ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ô LO160000	⌐ SM680000	╣ SF230000	╔ SF390000	┘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	║ SF240000	╬ SF400000	┌ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	¢ SC040000	½ NF010000	╗ SF250000	╦ SF410000	█ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▬ SF570000	∞ SA450000	ⁿ LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	‗ SM100000	Ù LU140000	¾ NF050000	╜ SF270000	═ SF430000	▌ SF580000	ϕ GF010001	² ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	À LA140000	Û LU160000	« SP170000	╛ SF280000	╬ SF440000	▐ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	§ SM240000	ƒ SC070000	» SP180000	┐ SF030000	╧ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00863

Code Page 864

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	° SM190000	β GB010000	(RSP) SP300000	٠ ND100001	¢ SC040000	ذ AD470000	ـ SM860000	ّ AX100004
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	• SD630000	∞ SA450000	(SHY) SP320000	١ ND010001	ء AX300000	ر AR010000	ف AF010003	ً AX100000
-2	♪ SM930000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	• SA790000	ϕ GF010001	آ AA210002	٢ ND020001	آ AA210000	ز AZ010000	ق AQ010003	ن AN010000
-3	♫ SM910000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	√ SA800000	± SA020000	£ SC020000	٣ ND030001	أ AA310000	س AS010003	ك AK010003	ه AH010000
-4	☼ SM690000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	▒ SF150000	½ NF010000	¤ SC010000	٤ ND040001	ؤ AW310000	ش AS230003	ل AL010003	ة AH010004
-5	═ SF430000	§ SM240000	⅚ SM020007	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	├ SF100000	¼ NF040000	أ AA310002	٥ ND050001	ع AC470002	ص AS450003	م AM010003	ى AA020002
-6	╢ SF240000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	┤ SF110000	≈ SA700000		٦ ND060001	ئ AY310000	ض AD450003	ن AN010003	ي AY010002
-7	╬ SF440000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	┼ SF050000	« SP170000	€ SC200000	٧ ND070001	ا AA010000	ط AT450000	ه AH010003	غ AG310004
-8	╣ SF230000	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	┤ SF090000	» SP180000	ا AA010002	٨ ND080001	ب AB010003	ظ AZ450000	و AW010000	ق AQ010000
-9	╗ SF410000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	┬ SF060000	لأ AL320000	ب AB010000	٩ ND090001	ة AT020000	ع AC470003	ى AA020000	لآ AL220000
-A	╠ SF420000	→ SM310000	* SM040007	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	├ SF080000	لأ AL320003	ت AT010000	ف AF010000	ت AT010003	غ AG310003	ي AY010003	لآ AL220003
-B	╝ SF400000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	┴ SF070000		ث AT470000	؛ SP140007	ث AT470003	إ SM650000	ض AD450000	ل AL010000
-C	╜ SF250000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	┐ SF030000		، SP080007	س AS010000	ج AG230003	¬ SM660000	ح AC470004	ك AK010000
-D	╓ SF390000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	┌ SF010000	لا AL020000	ج AG230000	ش AS230000	ح AH450003	÷ SA060000	خ AG310002	ي AY010000
-E	╙ SF380000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	└ SF020000	لا AL020003	ح AH450000	ص AS450000	خ AH470003	× SA070000	غ AG310000	■ SM470000
-F	╛ SF260000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	┘ SF040000	، SM870000	خ AH470000	؟ SP150007	د AD010000	ع AC470000	م AM010000	

Code Page 865

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	Ç LC420000	É LE120000	á LA110000	░ SF140000	└ SF020000	╩ SF460000	α GA010000	≡ SA480000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	ü LU170000	æ LA510000	í LI110000	▒ SF150000	┴ SF070000	╧ SF470000	ß LS610000	± SA020000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	é LE110000	Æ LA520000	ó LO110000	▓ SF160000	┬ SF060000	╥ SF480000	Γ GG020000	≥ SA530000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	â LA150000	ô LO150000	ú LU110000	│ SF110000	├ SF080000	╙ SF490000	π GP010000	≤ SA520000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	ä LA170000	ö LO170000	ñ LN190000	┤ SF090000	─ SF100000	╘ SF500000	Σ GS020000	⌠ SS260000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	à LA130000	ò LO130000	Ñ LN200000	╡ SF190000	┼ SF050000	╒ SF510000	σ GS010000	⌡ SS270000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	å LA270000	û LU150000	ª SM210000	╢ SF200000	╞ SF360000	╓ SF520000	µ GM010000	÷ SA060000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	ç LC410000	ù LU130000	º SM200000	╖ SF210000	╟ SF370000	╫ SF530000	τ GT010000	≈ SA700000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	ê LE150000	ÿ LY170000	¿ SP160000	╕ SF220000	╚ SF380000	╪ SF540000	Φ GF020000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	ë LE170000	Ö LO180000	⌐ SM680000	╣ SF230000	╔ SF390000	┘ SF040000	Θ GT620000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	è LE130000	Ü LU180000	¬ SM660000	║ SF240000	╩ SF400000	┌ SF010000	Ω GO320000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	ï LI170000	ø LO610000	½ NF010000	╗ SF250000	╦ SF410000	█ SF610000	δ GD010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	î LI150000	£ SC020000	¼ NF040000	╝ SF260000	╠ SF420000	▬ SF570000	∞ SA450000	ⁿ LN011000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	ì LI130000	Ø LO620000	¡ SP030000	╜ SF270000	═ SF430000	▌ SF580000	ϕ GF010001	² ND021000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	Ä LA180000	₧ SC060000	« SP170000	╛ SF280000	╬ SF440000	▐ SF590000	ε GE010000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	Å LA280000	ƒ SC070000	¤ SC010000	┐ SF030000	╨ SF450000	▬ SF600000	∩ SA380000	(RSP) SP300000

Code Page 00865

Code Page 866

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		► SM590000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000	А KA020000	Р KR020000	а KA010000	░ SF140000	┌ SF020000	╥ SF460000	р KR010000	Ё KE180000
-1	☺ SS000000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000	Б KB020000	С KS020000	б KB010000	▒ SF150000	┬ SF070000	╤ SF470000	с KS010000	ё KE170000
-2	☻ SS010000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000	В KV020000	Т KT020000	в KV010000	▓ SF160000	├ SF060000	╥ SF480000	т KT010000	Є KE160000
-3	♥ SS020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000	Г KG020000	У KU020000	г KG010000	│ SF110000	┼ SF080000	╙ SF490000	у KU010000	є KE150000
-4	♦ SS030000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000	Д KD020000	Ф KF020000	д KD010000	┤ SF090000	─ SF100000	╘ SF500000	ф KF010000	Ї KI180000
-5	♣ SS040000	§ SM240000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	Е KE020000	Х KH020000	е KE010000	╡ SF190000	┼ SF050000	╒ SF510000	х KH010000	ї KI170000
-6	♠ SS050000	▬ SM700000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000	Ж KZ220000	Ц KC020000	ж KZ210000	╢ SF200000	╞ SF360000	╒ SF520000	ц KC010000	Ў KU240000
-7	• SM570000	↨ SM770000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000	З KZ020000	Ч KC220000	з KZ010000	╖ SF210000	╟ SF370000	╫ SF530000	ч KC210000	ў KU230000
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	И KI020000	Ш KS220000	и KI010000	╕ SF220000	╚ SF380000	╪ SF540000	ш KS210000	° SM190000
-9	○ SM750000	↓ SM330000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000	Й KJ120000	Щ KS160000	й KJ110000	╣ SF230000	╔ SF390000	╘ SF040000	щ KS150000	• SA790000
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000	К KK020000	Ъ KU220000	к KK010000	╗ SF240000	╩ SF400000	╒ SF010000	ъ KU210000	· SD630000
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000	Л KL020000	Ы KY020000	л KL010000	╖ SF250000	╦ SF410000	▄ SF610000	ы KY010000	√ SA800000
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	М KM020000	Ь KX120000	м KM010000	╝ SF260000	╠ SF420000	▬ SF570000	ь KX110000	№ SM000000
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	Н KN020000	Э KE140000	н KN010000	╜ SF270000	═ SF430000	▌ SF580000	э KE130000	¤ SC010000
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000	О KO020000	Ю KU160000	о KO010000	╛ SF280000	╬ SF440000	▐ SF590000	ю KU150000	■ SM470000
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000	П KP020000	Я KA160000	п KP010000	┐ SF030000	═ SF450000	▬ SF600000	я KA150000	(RSP) SP300000

Code Page 869

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►	(SP)	0	@	P	`	p		΄Ι	ϊ	░	└	Τ	ζ	(SHY)
-1	☺	◄	!	1	A	Q	a	q		Ϊ	΄ϊ	▒	┴	Υ	η	±
-2	☻	↕	"	2	B	R	b	r		΄Ο	ό	▓	┬	Φ	θ	υ
-3	♥	!!	#	3	C	S	c	s			ύ	│	├	Χ	ι	φ
-4	♦	¶	$	4	D	T	d	t			Α	┤	─	Ψ	κ	χ
-5	♣	§	%	5	E	U	e	u		΄Υ	Β	Κ	┼	Ω	λ	§
-6	♠	▬	&	6	F	V	f	v	΄Α	Ϋ	Γ	Λ	Π	α	μ	ψ
-7	•	↨	'	7	G	W	g	w	€	©	Δ	Μ	Ρ	β	ν	΄
-8	◘	↑	(	8	H	X	h	x	·	΄Ω	Ε	Ν	╚	γ	ξ	°
-9	○	↓	)	9	I	Y	i	y	¬	²	Ζ	╝	╔	┘	ο	¨
-A	◙	→	*	:	J	Z	j	z	¦	³	Η	│	╩	┌	π	ω
-B	♂	←	+	;	K	[	k	{	‘	ά	½	╗	╦	█	ρ	ϋ
-C	♀	∟	,	<	L	\	l	|	’	£	Θ	╜	╠	▬	σ	΄ύ
-D	♪	↔	-	=	M	]	m	}	΄Ε	έ	Ι	Ξ	═	δ	ς	ώ
-E	♫	▲	.	>	N	^	n	~	—	ή	«	Ο	╬	ε	τ	■
-F	☼	▼	/	?	O	_	o	⌂	΄Η	ί	»	┐	Σ	▬	΄	(RSP)

Code Page 00869

Code Page 897

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		╬ SF440000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000				ー JX700000	タ JT100000	ミ JM200000		
-1	╔ SF390000		! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000			。 JQ700000	ア JA000000	チ JT200000	ム JM300000		
-2	╗ SF250000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000			「 JQ710000	イ JI000000	ツ JT300000	メ JM400000		
-3	╚ SF380000		# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000			」 JQ720000	ウ JU000000	テ JT400000	モ JM500000		
-4	╝ SF260000	▒ SF160000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000			、 JQ730000	エ JE000000	ト JT500000	ヤ JY100000		
-5	╢ SF240000	╩ SF400000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000			・ JQ740000	オ JO000000	ナ JN100000	ユ JY300000		
-6	═ SF430000	╠ SF410000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000			ヲ JW500000	カ JK100000	ニ JN200000	ヨ JY500000		
-7	↓ SM330000	╣ SF230000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000			ァ JA010000	キ JK200000	ヌ JN300000	ラ JR100000		
-8			(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000			ィ JI010000	ク JK300000	ネ JN400000	リ JR200000		
-9	○ SM750000	╢ SF420000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000			ゥ JU010000	ケ JK400000	ノ JN500000	ル JR300000		
-A		░ SF140000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000			ェ JE010000	コ JK500000	ハ JH100000	レ JR400000		
-B	⊠ SP500000	↵ SM720000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000			ォ JO010000	サ JS100000	ヒ JH200000	ロ JR500000		
-C		↑ SM320000	, SP080000	< SA030000	L LL020000	¥ SC050000	l LL010000	| SM130000			ャ JY110000	シ JS200000	フ JH300000	ワ JW100000		
-D		╓ SF110000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000			ュ JY310000	ス JS300000	ヘ JH400000	ン JN000000		
-E	■ SM470000	→ SM310000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	‾ SM150000			ョ JY510000	セ JS400000	ホ JH500000	゛ JX710000		
-F	☼ SM690000	← SM300000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000				ッ JT310000	ソ JS500000	マ JM100000	゜ JX720000		

Code Page 00897

Code Page 951

A1A1	A1A2	A1A3	A1A4	A1A5	A1A6	A1A7	A1A8	A1A9	A1AA	A1AB	A1AC	A1AD	A1AE	A1AF	A1B0
A1B1	A1B2	A1B3	A1B4	A1B5	A1B6	A1B7	A1B8	A1B9	A1BA	A1BB	A1BC	A1BD	A1BE	A1BF	A1C0
A1C1	A1C2	A1C3	A1C4	A1C5	A1C6	A1C7	A1C8	A1C9	A1CA	A1CB	A1CC	A1CD	A1D7	A1D8	A1D9
A1DA	A1DB	A1DC	A1DD	A1DE	A1DF	A1E0	A1E1	A1E2	A1E3	A1E4	A1E5	A1E6	A1E7	A1E8	A1E9
A1EA	A1EB	A3A1	A3A2	A3A3	A3A4	A3A5	A3A6	A3A7	A3A8	A3A9	A3AA	A3AB	A3AC	A3AD	A3AE
A3AF	A3B0	A3B1	A3B2	A3B3	A3B4	A3B5	A3B6	A3B7	A3B8	A3B9	A3BA	A3BB	A3BC	A3BD	A3BE
A3BF	A3C0	A3C1	A3C2	A3C3	A3C4	A3C5	A3C6	A3C7	A3C8	A3C9	A3CA	A3CB	A3CC	A3CD	A3CE
A3CF	A3D0	A3D1	A3D2	A3D3	A3D4	A3D5	A3D6	A3D7	A3D8	A3D9	A3DA	A3DB	A3DC	A3DD	A3DE
A3DF	A3E0	A3E1	A3E2	A3E3	A3E4	A3E5	A3E6	A3E7	A3E8	A3E9	A3EA	A3EB	A3EC	A3ED	A3EE
A3EF	A3F0	A3F1	A3F2	A3F3	A3F4	A3F5	A3F6	A3F7	A3F8	A3F9	A3FA	A3FB	A3FC	A3FD	A3FE
A4A1	A4A2	A4A3	A4A4	A4A5	A4A6	A4A7	A4A8	A4A9	A4AA	A4AB	A4AC	A4AD	A4AE	A4AF	A4B0
A4B1	A4B2	A4B3	A4B4	A4B5	A4B6	A4B7	A4B8	A4B9	A4BA	A4BB	A4BC	A4BD	A4BE	A4BF	A4C0
A4C1	A4C2	A4C3	A4C4	A4C5	A4C6	A4C7	A4C8	A4C9	A4CA	A4CB	A4CC	A4CD	A4CE	A4CF	A4D0
A4D1	A4D2	A4D3	AAA1	AAA2	AAA3	AAA4	AAA5	AAA6	AAA7	AAA8	AAA9	AAAA	AAAB	AAAC	AAAD
AAAE	AAAF	AAB0	AAB1	AAB2	AAB3	AAB4	AAB5	AAB6	AAB7	AAB8	AAB9	AABA	AABB	AABC	AABD
AABE	AABF	AAC0	AAC1	AAC2	AAC3	AAC4	AAC5	AAC6	AAC7	AAC8	AAC9	AACA	AACB	AACC	AACD

Figure 75. Special Symbols Alphanumeric/Numerics/Jamo/Hiragana/Hanja (1 of 3)

AACE	AACF	AAD0	AAD1	AAD2	AAD3	AAD4	AAD5	AAD6	AAD7	AAD8	AAD9	AADA	AADB	AADC	AADD
の	は	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ
AADE	AADF	AAE0	AAE1	AAE2	AAE3	AAE4	AAE5	AAE6	AAE7	AAE8	AAE9	AAEA	AAEB	AAEC	AAED
ま	み	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ
AAEE	AAEF	AAF0	AAF1	AAF2	AAF3	ABA1	ABA2	ABA3	ABA4	ABA5	ABA6	ABA7	ABA8	ABA9	ABAA
ゎ	わ	ゐ	ゑ	を	ん	ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ
ABAB	ABAC	ABAD	ABAE	ABAF	ABB0	ABB1	ABB2	ABB3	ABB4	ABB5	ABB6	ABB7	ABB8	ABB9	ABBA
カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ
ABBB	ABBC	ABBD	ABBE	ABBF	ABC0	ABC1	ABC2	ABC3	ABC4	ABC5	ABC6	ABC7	ABC8	ABC9	ABCA
セ	ゼ	ソ	ゾ	タ	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ
ABCB	ABCC	ABCD	ABCE	ABCF	ABD0	ABD1	ABD2	ABD3	ABD4	ABD5	ABD6	ABD7	ABD8	ABD9	ABDA
ニ	ヌ	ネ	ノ	ハ	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ
ABDB	ABDC	ABDD	ABDE	ABDF	ABE0	ABE1	ABE2	ABE3	ABE4	ABE5	ABE6	ABE7	ABE8	ABE9	ABEA
ホ	ボ	ポ	マ	ミ	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ
ABEB	ABEC	ABED	ABEE	ABEF	ABF0	ABF1	ABF2	ABF3	ABF4	ABF5	ABF6	A5B0	A5B1	A5B2	A5B3
ル	レ	ロ	ヮ	ワ	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ	Ⅰ	Ⅱ	Ⅲ	Ⅳ
A5B4	A5B5	A5B6	A5B7	A5B8	A5B9	A5C1	A5C2	A5C3	A5C4	A5C5	A5C6	A5C7	A5C8	A5C9	A5CA
Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ
A5CB	A5CC	A5CD	A5CE	A5CF	A5D0	A5D1	A5D2	A5D3	A5D4	A5D5	A5D6	A5D7	A5D8	A5E1	A5E2
Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	α	β
A5E3	A5E4	A5E5	A5E6	A5E7	A5E8	A5E9	A5EA	A5EB	A5EC	A5ED	A5EE	A5EF	A5F0	A5F1	A5F2
γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ
A5F3	A5F4	A5F5	A5F6	A5F7	A5F8	C3A3	CAA4	CAAB	CAE0	CBC1	CBC7	CBD2	CBD4	CBDB	CBEC
τ	υ	φ	χ	ψ	ω	仮	価	家	間	個	改	開	客	去	件
CBFE	CCAB	CCBD	CCBF	CCDA	CCF8	CCF9	CDAA	CDD3	CDD4	CDEA	CDEB	CDEC	CDFD	CEA1	CEDF
検	格	決	結	更	契	季	計	顧	高	供	公	共	果	科	交
CEFA	CEFD	CFA1	CFD1	CFD8	CFDB	CFE7	D0B3	D0C3	D0DD	D0EA	D0EC	D0F1	D1A2	D1C0	D1F5
九	具	区	局	群	郡	券	均	勤	金	企	其	基	期	記	南
D2A1	D2AE	D2B4	D3A4	D3B9	D3BD	D3DB	D3DE	D3E8	D3F8	D4CF	D4D4	D4F4	D4F5	D5D1	D5D7
納	内	年	単	達	担	代	大	貸	度	冬	東	登	等	来	両
D5E1	D5F4	D6E2	D6F5	D6F9	D7BE	D7BF	D7D7	D8B2	D7C7	D8DD	D8E2	D8E3	D8FC	D9A3	D9CA
量	力	録	了	料	類	六	利	万	末	枚	買	売	面	名	木

Figure 76. Special Symbols Alphanumeric/Numerics/Jamo/Hiragana/Hanja (2 of 3)

D9CD	D9ED	D9FE	DAA6	D9FD	DAAA	DAB1	DAE2	DAE3	DAF5	DAF7	DBA1	DBB0	DBC3	DBD5	DBDD
目	無	文	門	問	物	未	半	反	般	返	発	方	倍	配	百
DBE3	DBF6	DCA8	DCAC	DCC3	DCD7	DDBB	DDC2	DDD5	DDD7	DDE1	DEA8	DEAA	DEBC	DEC0	DEC5
番	法	変	別	報	服	部	分	不	払	備	費	非	氷	事	使
DECC	DEDB	DEE4	DEE7	DFA7	DFA9	DFB2	DFBE	DFC2	DFE6	E0B4	E0BB	E0E2	E0F7	E1AA	E1B3
四	査	社	私	産	算	三	上	商	生	石	先	設	成	税	小
E1B4	E1B6	E1BC	E1EA	E1F4	E2A2	E2A5	E2A6	E2A9	E2ED	E2F7	E3A7	E3AF	E3B7	E3BC	E3C1
少	所	消	送	受	手	収	数	水	純	順	習	承	始	市	時
E3CB	E3D2	E3E1	E3E6	E3E9	E3F7	E4A8	E4CC	E4D0	E4E7	E4FE	E5B3	E5D5	E5EB	E5F6	E6B6
試	式	信	新	申	失	十	安	案	央	額	約	陽	言	業	易
E7BD	E7E8	E7E9	E7ED	E8A6	E8C7	E8D9	E8DD	E8E2	E9C4	E9D3	EAAA	EAAB	EAAC	EAC5	EAF3
営	預	五	午	誤	完	往	王	外	用	右	元	原	自	月	有
EBBF	EBDE	EBFD	ECA3	ECA4	ECCC	ECD1	ECDA	ECE3	ECE9	ECED	ECFD	ECAD	EDAE	EDBA	EDC0
肉	銀	衣	二	以	益	人	引	認	一	日	入	子	字	者	資
EDC2	EDD1	EDDE	EDE5	EEA2	EEA4	EEA7	EEDD	EEEF	EEF1	EFB1	EFB7	EFC1	EFC3	EFD2	EFE1
作	残	場	張	再	在	材	積	全	前	銭	切	店	点	定	正
EFF1	EFF4	F0A4	F0FA	F0FB	F1A7	F1AC	F1BB	F1CE	F1DE	F1E9	F2A4	F2A5	F2A8	F2AD	F3AC
精	訂	制	種	絡	左	住	株	週	準	中	志	持	支	止	差
F4A1	F4B6	F4F0	F4F8	F5C5	F5D5	F5E6	F5F0	F5F3	F6A2	F6B7	F6D2	F6E2	F7CF	F7D7	F7E5
責	千	替	初	総	秋	祝	春	出	取	値	七	他	土	通	特
F7FC	F8A1	F8A2	F8C1	F8CD	F8F6	F8F9	F8FA	F9A1	F9BB	F9BE	F9C3	F9D3	F9DC	F9EA	FABC
販	阪	八	平	閉	標	票	表	品	下	夏	荷	漢	割	合	行
FADE	FAF0	FBA1	FBBC	FBDC	FBFD	FCB5	FCC0	FCDE	FCE5	FDAD	FDBA	FDCC			
現	協	形	呼	号	火	換	活	回	会	後	訓	休			

Figure 77. Special Symbols Alphanumeric/Numerics/Jamo/Hiragana/Hanja (3 of 3)

B0A1	B0A2	B0A3	B0A4	B0A5	B0A6	B0A7	B0A8	B0A9	B0AA	B0AB	B0AC	B0AD	B0AE	B0AF	B0B0
가	각	간	갇	갈	갉	갊	감	갑	값	갓	갔	강	갖	갗	갘
B0B1	B0B2	B0B3	B0B4	B0B5	B0B6	B0B7	B0B8	B0B9	B0BA	B0BB	B0BC	B0BD	B0BE	B0BF	B0C0
갚	갛	개	객	갠	갤	갬	갭	갯	갰	갱	갸	갹	갼	걀	걋
B0C1	B0C2	B0C3	B0C4	B0C5	B0C6	B0C7	B0C8	B0C9	B0CA	B0CB	B0CC	B0CD	B0CE	B0CF	B0D0
걍	걔	걘	걜	거	걱	건	걷	걸	걺	검	겁	것	겄	겅	겆
B0D1	B0D2	B0D3	B0D4	B0D5	B0D6	B0D7	B0D8	B0D9	B0DA	B0DB	B0DC	B0DD	B0DE	B0DF	B0E0
겉	겊	겋	게	겐	겔	겜	겝	겟	겠	겡	겨	격	겪	견	겯
B0E1	B0E2	B0E3	B0E4	B0E5	B0E6	B0E7	B0E8	B0E9	B0EA	B0EB	B0EC	B0ED	B0EE	B0EF	B0F0
결	겸	겹	겻	겼	경	곁	계	곈	곌	곕	곗	고	곡	곤	곧
B0F1	B0F2	B0F3	B0F4	B0F5	B0F6	B0F7	B0F8	B0F9	B0FA	B0FB	B0FC	B0FD	B0FE	B1A1	B1A2
골	곪	곬	곯	곰	곱	곳	공	곶	과	곽	관	괄	괄	괌	괍
B1A3	B1A4	B1A5	B1A6	B1A7	B1A8	B1A9	B1AA	B1AB	B1AC	B1AD	B1AE	B1AF	B1B0	B1B1	B1B2
괏	광	괘	괜	괠	괩	괬	괭	괴	괵	괸	괼	굄	굅	굇	굉
B1B3	B1B4	B1B5	B1B6	B1B7	B1B8	B1B9	B1BA	B1BB	B1BC	B1BD	B1BE	B1BF	B1C0	B1C1	B1C2
교	굔	굘	굡	굣	구	국	군	굳	굴	굵	굶	굷	굼	굽	굿
B1C3	B1C4	B1C5	B1C6	B1C7	B1C8	B1C9	B1CA	B1CB	B1CC	B1CD	B1CE	B1CF	B1D0	B1D1	B1D2
궁	궂	궈	궉	권	궐	궜	궝	궤	궷	귀	귁	귄	귈	귐	귑
B1D3	B1D4	B1D5	B1D6	B1D7	B1D8	B1D9	B1DA	B1DB	B1DC	B1DD	B1DE	B1DF	B1E0	B1E1	B1E2
귓	그	귿	글	긁	금	급	긋	긍	긎	긍	긋	긍	긋	긔	기
B1E3	B1E4	B1E5	B1E6	B1E7	B1E8	B1E9	B1EA	B1EB	B1EC	B1ED	B1EE	B1EF	B1F0	B1F1	B1F2
긱	긴	길	긷	길	김	깁	깃	깅	깆	깊	까	깍	깎	깐	깔
B1F3	B1F4	B1F5	B1F6	B1F7	B1F8	B1F9	B1FA	B1FB	B1FC	B1FD	B1FE	B2A1	B2A2	B2A3	B2A4
깖	깜	깝	깟	깠	깡	깥	깨	깩	깬	깰	깸	깹	깻	깼	깽
B2A5	B2A6	B2A7	B2A8	B2A9	B2AA	B2AB	B2AC	B2AD	B2AE	B2AF	B2B0	B2B1	B2B2	B2B3	B2B4
꺄	꺅	꺌	꺼	꺽	꺾	껀	껄	껌	껍	껏	껐	껑	께	껙	껜
B2B5	B2B6	B2B7	B2B8	B2B9	B2BA	B2BB	B2BC	B2BD	B2BE	B2BF	B2C0	B2C1	B2C2	B2C3	B2C4
껠	껨	껩	껴	껸	껼	껴	껼	껼	께	꺼	꼭	꼰	꼲	꼴	꼼
B2C5	B2C6	B2C7	B2C8	B2C9	B2CA	B2CB	B2CC	B2CD	B2CE	B2CF	B2D0	B2D1	B2D2	B2D3	B2D4
꼽	꼿	꽁	꽃	꽈	과	곽	괄	괄	광	괘	괙	괭	괴	괸	괼
B2D5	B2D6	B2D7	B2D8	B2D9	B2DA	B2DB	B2DC	B2DD	B2DE	B2DF	B2E0	B2E1	B2E2	B2E3	B2E4
굄	굅	굉	교	구	국	군	굴	굵	굼	굽	굿	궁	궂	궈	궐

Figure 78. Hangeul (1 of 10)

B2E5	B2E6	B2E7	B2E8	B2E9	B2EA	B2EB	B2EC	B2ED	B2EE	B2EF	B2F0	B2F1	B2F2	B2F3	B2F4
쩐	쩐	쩨	쩨	쩬	쩰	쩸	쩹	쩼	쩌	쩐	쩔	쩜	쩝	쩌	쩐
B2F5	B2F6	B2F7	B2F8	B2F9	B2FA	B2FB	B2FC	B2FD	B2FE	B3A1	B3A2	B3A3	B3A4	B3A5	B3A6
쩐	쩐	쩐	쩔	쩔	쩔	쩜	쩝	쩌	쩐	쩔	찌	찌	찌	찌	찌
B3A7	B3A8	B3A9	B3AA	B3AB	B3AC	B3AD	B3AE	B3AF	B3B0	B3B1	B3B2	B3B3	B3B4	B3B5	B3B6
찌	찌	찌	나	나	나	나	나	날	날	날	남	남	나	나	나
B3B7	B3B8	B3B9	B3BA	B3BB	B3BC	B3BD	B3BE	B3BF	B3C0	B3C1	B3C2	B3C3	B3C4	B3C5	B3C6
나	나	날	나	내	내	내	내	내	내	내	내	내	냐	냐	냐
B3C7	B3C8	B3C9	B3CA	B3CB	B3CC	B3CD	B3CE	B3CF	B3D0	B3D1	B3D2	B3D3	B3D4	B3D5	B3D6
냘	냠	냥	너	너	너	너	너	너	너	너	너	너	너	너	너
B3D7	B3D8	B3D9	B3DA	B3DB	B3DC	B3DD	B3DE	B3DF	B3E0	B3E1	B3E2	B3E3	B3E4	B3E5	B3E6
네	네	네	넬	넴	넵	넷	넸	넹	녀	녀	녀	녈	녀	녀	녀
B3E7	B3E8	B3E9	B3EA	B3EB	B3EC	B3ED	B3EE	B3EF	B3F0	B3F1	B3F2	B3F3	B3F4	B3F5	B3F6
녕	녘	녜	녠	노	녹	논	놀	놀	놈	놉	놋	농	높	놓	놔
B3F7	B3F8	B3F9	B3FA	B3FB	B3FC	B3FD	B3FE	B4A1	B4A2	B4A3	B4A4	B4A5	B4A6	B4A7	B4A8
놘	놜	놨	뇌	뇐	뇔	뇜	뇝	뇟	뇨	뇩	뇬	뇰	뇹	뇻	뇽
B4A9	B4AA	B4AB	B4AC	B4AD	B4AE	B4AF	B4B0	B4B1	B4B2	B4B3	B4B4	B4B5	B4B6	B4B7	B4B8
누	눅	눈	눋	눌	눔	눕	눗	눙	뉘	뉚	뉘	뉘	뉜	뉠	뉨
B4B9	B4BA	B4BB	B4BC	B4BD	B4BE	B4BF	B4C0	B4C1	B4C2	B4C3	B4C4	B4C5	B4C6	B4C7	B4C8
뉩	뉴	뉵	뉼	뉽	뉾	뉿	느	늑	는	늘	늘	늠	늡	늣	능
B4C9	B4CA	B4CB	B4CC	B4CD	B4CE	B4CF	B4D0	B4D1	B4D2	B4D3	B4D4	B4D5	B4D6	B4D7	B4D8
능	늦	늧	니	닌	닐	니	닉	닌	닐	닒	님	닙	닛	닝	닢
B4D9	B4DA	B4DB	B4DC	B4DD	B4DE	B4DF	B4E0	B4E1	B4E2	B4E3	B4E4	B4E5	B4E6	B4E7	B4E8
다	닥	닫	단	닫	달	닭	닮	닯	닳	담	답	닷	닸	당	닺
B4E9	B4EA	B4EB	B4EC	B4ED	B4EE	B4EF	B4F0	B4F1	B4F2	B4F3	B4F4	B4F5	B4F6	B4F7	B4F8
닾	닿	대	댁	댄	댈	댐	댑	댓	댔	댕	댜	더	덕	덖	던
B4F9	B4FA	B4FB	B4FC	B4FD	B4FE	B5A1	B5A2	B5A3	B5A4	B5A5	B5A6	B5A7	B5A8	B5A9	B5AA
덛	덜	덞	덟	덤	덥	덧	덩	덫	덮	데	덱	덴	델	뎀	뎁
B5AB	B5AC	B5AD	B5AE	B5AF	B5B0	B5B1	B5B2	B5B3	B5B4	B5B5	B5B6	B5B7	B5B8	B5B9	B5BA
뎃	뎄	뎅	더	덛	덜	덞	덟	뎨	뎬	도	독	돈	돋	돌	돐
B5BB	B5BC	B5BD	B5BE	B5BF	B5C0	B5C1	B5C2	B5C3	B5C4	B5C5	B5C6	B5C7	B5C8	B5C9	B5CA
돔	돔	돕	돗	동	돛	돝	돠	돤	돨	돼	됐	되	된	될	됨

Figure 79. Hangeul (2 of 10)

B5CB	B5CC	B5CD	B5CE	B5CF	B5D0	B5D1	B5D2	B5D3	B5D4	B5D5	B5D6	B5D7	B5D8	B5D9	B5DA
뒵	되	둡	두	둑	둔	둘	둠	둡	둣	둥	뒤	뒀	뒈	뒝	뒤
B5DB	B5DC	B5DD	B5DE	B5DF	B5E0	B5E1	B5E2	B5E3	B5E4	B5E5	B5E6	B5E7	B5E8	B5E9	B5EA
뒨	뒬	뒵	뒷	뒹	듀	듄	듈	듐	듕	드	득	든	들	글	듦
B5EB	B5EC	B5ED	B5EE	B5EF	B5F0	B5F1	B5F2	B5F3	B5F4	B5F5	B5F6	B5F7	B5F8	B5F9	B5FA
듬	듭	듯	등	듸	디	딕	딘	딛	딜	딤	딥	딧	딨	딩	디
B5FB	B5FC	B5FD	B5FE	B6A1	B6A2	B6A3	B6A4	B6A5	B6A6	B6A7	B6A8	B6A9	B6AA	B6AB	B6AC
따	딱	딴	딸	땀	땁	땃	땄	땅	땆	때	땍	땐	땔	땜	땝
B6AD	B6AE	B6AF	B6B0	B6B1	B6B2	B6B3	B6B4	B6B5	B6B6	B6B7	B6B8	B6B9	B6BA	B6BB	B6BC
땟	땠	땡	떠	떡	떤	떨	떪	떬	떰	떱	떳	떴	떵	떻	떼
B6BD	B6BE	B6BF	B6C0	B6C1	B6C2	B6C3	B6C4	B6C5	B6C6	B6C7	B6C8	B6C9	B6CA	B6CB	B6CC
떽	떤	떨	떰	떱	떳	떴	떵	떠	떡	또	똑	똔	똘	똥	똬
B6CD	B6CE	B6CF	B6D0	B6D1	B6D2	B6D3	B6D4	B6D5	B6D6	B6D7	B6D8	B6D9	B6DA	B6DB	B6DC
뙬	뙈	뙤	뙨	뚀	뚜	뚝	뚠	뚤	뚬	뚱	뛔	뛰	뛴	뛸	뜀
B6DD	B6DE	B6DF	B6E0	B6E1	B6E2	B6E3	B6E4	B6E5	B6E6	B6E7	B6E8	B6E9	B6EA	B6EB	B6EC
뜁	뜅	뜨	뜩	뜬	뜯	뜰	뜸	뜹	뜻	띄	띈	띌	띔	띕	띠
B6ED	B6EE	B6EF	B6F0	B6F1	B6F2	B6F3	B6F4	B6F5	B6F6	B6F7	B6F8	B6F9	B6FA	B6FB	B6FC
띤	띨	띰	띱	띳	띵	라	락	란	랄	람	랍	랏	랐	랑	랒
B6FD	B6FE	B7A1	B7A2	B7A3	B7A4	B7A5	B7A6	B7A7	B7A8	B7A9	B7AA	B7AB	B7AC	B7AD	B7AE
랖	랗	래	랙	랜	랠	램	랩	랫	랬	랭	랴	략	랸	럇	량
B7AF	B7B0	B7B1	B7B2	B7B3	B7B4	B7B5	B7B6	B7B7	B7B8	B7B9	B7BA	B7BB	B7BC	B7BD	B7BE
러	럭	런	럴	럼	럽	럿	렀	렁	렂	레	렉	렌	렐	렘	렙
B7BF	B7C0	B7C1	B7C2	B7C3	B7C4	B7C5	B7C6	B7C7	B7C8	B7C9	B7CA	B7CB	B7CC	B7CD	B7CE
렛	렝	려	력	련	렬	렴	렵	렷	렸	령	례	롄	롈	로	로
B7CF	B7D0	B7D1	B7D2	B7D3	B7D4	B7D5	B7D6	B7D7	B7D8	B7D9	B7DA	B7DB	B7DC	B7DD	B7DE
록	론	롤	롬	롭	롯	롱	롸	롼	뢍	뢨	뢰	뢴	뢸	룀	뢉
B7DF	B7E0	B7E1	B7E2	B7E3	B7E4	B7E5	B7E6	B7E7	B7E8	B7E9	B7EA	B7EB	B7EC	B7ED	B7EE
뢋	뢍	료	룐	룔	룝	룟	룡	루	룩	룬	룰	룸	룹	룻	룽
B7EF	B7F0	B7F1	B7F2	B7F3	B7F4	B7F5	B7F6	B7F7	B7F8	B7F9	B7FA	B7FB	B7FC	B7FD	B7FE
뤼	뤘	뤠	뤼	뤽	뤤	뤨	뤼	뤽	뤼	르	륙	륜	률	륨	륩
B8A1	B8A2	B8A3	B8A4	B8A5	B8A6	B8A7	B8A8	B8A9	B8AA	B8AB	B8AC	B8AD	B8AE	B8AF	B8B0
륫	륭	르	릑	릔	릗	릚	릡	릣	릥	릦	릧	릩	리	리	릭

Figure 80. Hangeul (3 of 10)

B8B1	B8B2	B8B3	B8B4	B8B5	B8B6	B8B7	B8B8	B8B9	B8BA	B8BB	B8BC	B8BD	B8BE	B8BF	B8C0
맇	맊	맋	만	맍	마	막	만	많	맏	말	맗	맘	맙	맚	맛
B8C1	B8C2	B8C3	B8C4	B8C5	B8C6	B8C7	B8C8	B8C9	B8CA	B8CB	B8CC	B8CD	B8CE	B8CF	B8D0
망	맞	맡	맢	매	맥	맨	맫	맬	맭	맮	맯	맰	맱	맲	맳
B8D1	B8D2	B8D3	B8D4	B8D5	B8D6	B8D7	B8D8	B8D9	B8DA	B8DB	B8DC	B8DD	B8DE	B8DF	B8E0
맴	맵	머	먹	먼	멀	멂	멈	멉	멊	멋	멌	멍	메	멕	멘
B8E1	B8E2	B8E3	B8E4	B8E5	B8E6	B8E7	B8E8	B8E9	B8EA	B8EB	B8EC	B8ED	B8EE	B8EF	B8F0
멜	멤	멥	멧	멨	멩	며	멱	면	멸	몃	몄	명	몇	몌	모
B8F1	B8F2	B8F3	B8F4	B8F5	B8F6	B8F7	B8F8	B8F9	B8FA	B8FB	B8FC	B8FD	B8FE	B9A1	B9A2
목	몫	몬	몰	몲	몸	몹	못	몽	뫄	뫈	뫘	뫙	뫼	묀	묄
B9A3	B9A4	B9A5	B9A6	B9A7	B9A8	B9A9	B9AA	B9AB	B9AC	B9AD	B9AE	B9AF	B9B0	B9B1	B9B2
묍	묏	묑	묘	묜	묠	묩	묫	무	묵	묶	문	묻	물	묽	몫
B9B3	B9B4	B9B5	B9B6	B9B7	B9B8	B9B9	B9BA	B9BB	B9BC	B9BD	B9BE	B9BF	B9C0	B9C1	B9C2
묾	묿	뭇	뭉	뭍	뭏	뭐	뭔	뭘	뭡	뭣	뭬	뮈	뮌	뮐	뮤
B9C3	B9C4	B9C5	B9C6	B9C7	B9C8	B9C9	B9CA	B9CB	B9CC	B9CD	B9CE	B9CF	B9D0	B9D1	B9D2
뮨	뮬	뮴	뮷	므	믄	믈	믐	믓	미	믹	민	믿	밀	밂	밈
B9D3	B9D4	B9D5	B9D6	B9D7	B9D8	B9D9	B9DA	B9DB	B9DC	B9DD	B9DE	B9DF	B9E0	B9E1	B9E2
밉	밋	밌	밍	밎	및	바	박	밖	밗	반	받	발	밝	밟	밟
B9E3	B9E4	B9E5	B9E6	B9E7	B9E8	B9E9	B9EA	B9EB	B9EC	B9ED	B9EE	B9EF	B9F0	B9F1	B9F2
밖	밥	밧	방	밭	배	백	밴	밸	뱀	뱁	뱃	뱄	뱅	뱉	뱌
B9F3	B9F4	B9F5	B9F6	B9F7	B9F8	B9F9	B9FA	B9FB	B9FC	B9FD	B9FE	BAA1	BAA2	BAA3	BAA4
뱍	뱐	뱝	버	벅	번	벋	벌	벎	범	법	벗	벙	벚	베	벡
BAA5	BAA6	BAA7	BAA8	BAA9	BAAA	BAAB	BAAC	BAAD	BAAE	BAAF	BAB0	BAB1	BAB2	BAB3	BAB4
벤	벧	벨	벩	벱	벳	벵	벼	벽	변	별	볍	볏	볐	병	볕
BAB5	BAB6	BAB7	BAB8	BAB9	BABA	BABB	BABC	BABD	BABE	BABF	BAC0	BAC1	BAC2	BAC3	BAC4
볕	볘	볜	보	복	볶	본	볼	봄	봅	봇	봉	봐	봔	봤	봬
BAC5	BAC6	BAC7	BAC8	BAC9	BACA	BACB	BACC	BACD	BACE	BACF	BAD0	BAD1	BAD2	BAD3	BAD4
뵀	뵈	뵉	뵌	뵐	뵘	뵙	뵤	뵨	부	북	분	불	붉	붊	붐
BAD5	BAD6	BAD7	BAD8	BAD9	BADA	BADB	BADC	BADD	BADE	BADF	BAE0	BAE1	BAE2	BAE3	BAE4
붑	붓	붕	붙	붇	붚	뷔	뷘	뷜	뷔	뷔	뷕	뷘	뷜	뷩	뷰
BAE5	BAE6	BAE7	BAE8	BAE9	BAEA	BAEB	BAEC	BAED	BAEE	BAEF	BAF0	BAF1	BAF2	BAF3	BAF4
뷴	뷸	뷤	뷧	뷩	븨	븍	븐	블	븜	븝	븟	비	빅	빈	빌

Figure 81. Hangeul (4 of 10)

BAF5	BAF6	BAF7	BAF8	BAF9	BAFA	BAFB	BAFC	BAFD	BAFE	BBA1	BBA2	BBA3	BBA4	BBA5	BBA6
뷔	뷕	뷘	뷔	뷕	뷔	뷕	빠	빡	빡	빨	빨	빠	빱	빠	빠
BBA7	BBA8	BBA9	BBAA	BBAB	BBAC	BBAD	BBAE	BBAF	BBB0	BBB1	BBB2	BBB3	BBB4	BBB5	BBB6
빠	빵	빼	빼	빼	빼	빼	빼	빼	빼	빼	빠	빠	빠	뻐	뻐
BBB7	BBB8	BBB9	BBBA	BBBB	BBBC	BBBD	BBBE	BBBF	BBC0	BBC1	BBC2	BBC3	BBC4	BBC5	BBC6
뻐	뻐	뻐	뻐	뻐	뻐	뻐	뻬	뻥	뻐	뻐	뻐	뻐	뻐	뻐	뻐
BBC7	BBC8	BBC9	BBCA	BBCB	BBCC	BBCD	BBCE	BBCF	BBD0	BBD1	BBD2	BBD3	BBD4	BBD5	BBD6
뽀	뽀	뽀	뽀	뽀	뽀	뽀	뾔	뽀	뽀	뽀	뽀	뽀	뽀	뽀	뽀
BBD7	BBD8	BBD9	BBDA	BBDB	BBDC	BBDD	BBDE	BBDF	BBE0	BBE1	BBE2	BBE3	BBE4	BBE5	BBE6
뽀	뽀	뽀	뽀	뽀	뽀	뽀	뽀	뾔	뾕	뾕	뾕	뾕	뾕	뾕	뾕
BBE7	BBE8	BBE9	BBEA	BBEB	BBEC	BBED	BBEE	BBEF	BBF0	BBF1	BBF2	BBF3	BBF4	BBF5	BBF6
사	삭	삭	삭	삭	삭	삭	삭	삭	삭	삭	삭	삭	삭	새	새
BBF7	BBF8	BBF9	BBFA	BBFB	BBFC	BBFD	BBFE	BCA1	BCA2	BCA3	BCA4	BCA5	BCA6	BCA7	BCA8
새	새	새	새	새	새	새	샤	샥	샥	샥	샥	샥	샥	샥	새
BCA9	BCAA	BCAB	BCAC	BCAD	BCAE	BCAF	BCB0	BCB1	BCB2	BCB3	BCB4	BCB5	BCB6	BCB7	BCB8
새	새	새	새	서	서	서	서	서	서	서	서	서	서	서	서
BCB9	BCBA	BCBB	BCBC	BCBD	BCBE	BCBF	BCC0	BCC1	BCC2	BCC3	BCC4	BCC5	BCC6	BCC7	BCC8
서	서	서	세	세	세	세	세	세	세	세	세	셔	셔	셔	셔
BCC9	BCCA	BCCB	BCCC	BCCD	BCCE	BCCF	BCD0	BCD1	BCD2	BCD3	BCD4	BCD5	BCD6	BCD7	BCD8
셔	셔	셔	셔	셔	셰	셰	셰	셰	소	속	속	손	솔	솔	솜
BCD9	BCDA	BCDB	BCDC	BCDD	BCDE	BCDF	BCE0	BCE1	BCE2	BCE3	BCE4	BCE5	BCE6	BCE7	BCE8
솜	소	송	솔	솨	솩	솬	솔	송	쇄	쇈	쇌	쇔	쇘	쇘	쇠
BCE9	BCEA	BCEB	BCEC	BCED	BCEE	BCEF	BCF0	BCF1	BCF2	BCF3	BCF4	BCF5	BCF6	BCF7	BCF8
쇤	쇨	쇰	쇱	쇳	쇼	쇽	쇤	쇨	쇰	쇱	쇳	쇵	쇽	쇤	쇤
BCF9	BCFA	BCFB	BCFC	BCFD	BCFE	BDA1	BDA2	BDA3	BDA4	BDA5	BDA6	BDA7	BDA8	BDA9	BDAA
쇤	쇨	쇰	쇱	쇳	쇵	쇽	숄	숑	숴	숴	쉐	쉐	쉘	쉐	쉠
BDAB	BDAC	BDAD	BDAE	BDAF	BDB0	BDB1	BDB2	BDB3	BDB4	BDB5	BDB6	BDB7	BDB8	BDB9	BDBA
쉐	쉬	쉬	쉰	쉴	쉼	쉽	쉿	쉬	슈	쉬	쉴	쉼	쉿	쉬	쉬
BDBB	BDBC	BDBD	BDBE	BDBF	BDC0	BDC1	BDC2	BDC3	BDC4	BDC5	BDC6	BDC7	BDC8	BDC9	BDCA
슈	슨	슬	슭	슴	습	슷	슝	시	신	신	신	실	싫	심	십
BDCB	BDCC	BDCD	BDCE	BDCF	BDD0	BDD1	BDD2	BDD3	BDD4	BDD5	BDD6	BDD7	BDD8	BDD9	BDDA
싯	싱	싸	싸	싸	싸	싸	싸	싸	싸	싸	싸	쌔	쌔	쌔	쌔

Figure 82. Hangeul (5 of 10)

BDDB	BDDC	BDDD	BDDE	BDDF	BDE0	BDE1	BDE2	BDE3	BDE4	BDE5	BDE6	BDE7	BDE8	BDE9	BDEA
쌜	쌤	쌥	쌧	쌩	쌪	써	썩	썬	썰	썲	썸	썹	썻	썽	쎄
BDEB	BDEC	BDED	BDEE	BDEF	BDF0	BDF1	BDF2	BDF3	BDF4	BDF5	BDF6	BDF7	BDF8	BDF9	BDFA
쎈	쎌	쎔	쎠	쏘	쏙	쏜	쏠	쏨	쏩	쏭	쏴	쏼	쏸	쏼	쐈
BDFB	BDFC	BDFD	BDFE	BEA1	BEA2	BEA3	BEA4	BEA5	BEA6	BEA7	BEA8	BEA9	BEAA	BEAB	BEAC
쐐	쐤	쐬	쐰	쐴	쐼	쐽	쑈	쑤	쑥	쑨	쑬	쑴	쑵	쑹	쒀
BEAD	BEAE	BEAF	BEB0	BEB1	BEB2	BEB3	BEB4	BEB5	BEB6	BEB7	BEB8	BEB9	BEBA	BEBB	BEBC
쒀	쒜	쒸	쒼	쓩	쓰	쓱	쓴	쓸	쓺	쓼	쓽	쓿	씨	씬	씰
BEBD	BEBE	BEBF	BEC0	BEC1	BEC2	BEC3	BEC4	BEC5	BEC6	BEC7	BEC8	BEC9	BECA	BECB	BECC
씸	씨	씩	씬	씰	씸	씹	씻	씽	아	악	안	앉	않	알	앍
BECD	BECE	BECF	BED0	BED1	BED2	BED3	BED4	BED5	BED6	BED7	BED8	BED9	BEDA	BEDB	BEDC
앎	앞	암	압	앗	았	앙	앝	앞	애	액	앤	앨	앰	앱	앳
BEDD	BEDE	BEDF	BEE0	BEE1	BEE2	BEE3	BEE4	BEE5	BEE6	BEE7	BEE8	BEE9	BEEA	BEEB	BEEC
앴	앵	야	약	얀	얄	얇	얌	얍	얏	양	얕	얗	얘	얜	얠
BEED	BEEE	BEEF	BEF0	BEF1	BEF2	BEF3	BEF4	BEF5	BEF6	BEF7	BEF8	BEF9	BEFA	BEFB	BEFC
얩	어	억	언	얻	어	억	얽	얾	어	얍	얐	얏	양	얒	얘
BEFD	BEFE	BFA1	BFA2	BFA3	BFA4	BFA5	BFA6	BFA7	BFA8	BFA9	BFAA	BFAB	BFAC	BFAD	BFAE
얙	얚	에	엑	엔	엘	엠	엡	엣	엥	여	역	엮	연	열	엶
BFAF	BFB0	BFB1	BFB2	BFB3	BFB4	BFB5	BFB6	BFB7	BFB8	BFB9	BFBA	BFBB	BFBC	BFBD	BFBE
엻	염	엽	엾	엿	였	영	옅	옆	옇	예	옌	옐	옘	옙	옛
BFBF	BFC0	BFC1	BFC2	BFC3	BFC4	BFC5	BFC6	BFC7	BFC8	BFC9	BFCA	BFCB	BFCC	BFCD	BFCE
옜	오	옥	온	올	옭	옴	옵	옷	옹	옺	옷	옹	옾	와	왁
BFCF	BFD0	BFD1	BFD2	BFD3	BFD4	BFD5	BFD6	BFD7	BFD8	BFD9	BFDA	BFDB	BFDC	BFDD	BFDE
완	왈	왐	왑	왓	왔	왕	왜	왝	왠	왤	왬	왭	외	왹	왠
BFDF	BFE0	BFE1	BFE2	BFE3	BFE4	BFE5	BFE6	BFE7	BFE8	BFE9	BFEA	BFEB	BFEC	BFED	BFEE
왤	왬	왭	왯	왱	요	욕	욘	욜	욤	욥	욧	용	우	욱	욘
BFEF	BFF0	BFF1	BFF2	BFF3	BFF4	BFF5	BFF6	BFF7	BFF8	BFF9	BFFA	BFFB	BFFC	BFFD	BFFE
울	욳	욶	우	욱	욷	욷	웃	웍	웍	웍	웍	웍	웍	웍	웨
C0A1	C0A2	C0A3	C0A4	C0A5	C0A6	C0A7	C0A8	C0A9	C0AA	C0AB	C0AC	C0AD	C0AE	C0AF	C0B0
웩	웬	웰	웸	웹	웻	위	윅	윈	윌	윔	윕	윗	윙	유	육
C0B1	C0B2	C0B3	C0B4	C0B5	C0B6	C0B7	C0B8	C0B9	C0BA	C0BB	C0BC	C0BD	C0BE	C0BF	C0C0
윤	율	윰	윱	윳	융	윷	으	윽	은	을	읊	음	읍	읏	응

Figure 83. Hangeul (6 of 10)

C0C1	C0C2	C0C3	C0C4	C0C5	C0C6	C0C7	C0C8	C0C9	C0CA	C0CB	C0CC	C0CD	C0CE	C0CF	C0D0
읖	읗	읙	읚	읛	읜	읝	읞	읟	읠	읡	이	익	인	일	읽
C0D1	C0D2	C0D3	C0D4	C0D5	C0D6	C0D7	C0D8	C0D9	C0DA	C0DB	C0DC	C0DD	C0DE	C0DF	C0E0
읾	잀	임	입	잇	있	잉	잊	잎	자	작	잔	잖	잗	잘	잚
C0E1	C0E2	C0E3	C0E4	C0E5	C0E6	C0E7	C0E8	C0E9	C0EA	C0EB	C0EC	C0ED	C0EE	C0EF	C0F0
잠	잡	잣	잤	장	잦	재	잭	잰	잴	잼	잽	잿	쟁	쟀	쟈
C0F1	C0F2	C0F3	C0F4	C0F5	C0F6	C0F7	C0F8	C0F9	C0FA	C0FB	C0FC	C0FD	C0FE	C1A1	C1A2
쟉	쟌	쟎	쟐	쟘	쟝	쟤	쟨	쟬	저	적	전	절	젊	점	접
C1A3	C1A4	C1A5	C1A6	C1A7	C1A8	C1A9	C1AA	C1AB	C1AC	C1AD	C1AE	C1AF	C1B0	C1B1	C1B2
젓	정	젖	졔	졔	젠	졜	졔	졘	졔	졔	져	젼	졀	졈	졉
C1B3	C1B4	C1B5	C1B6	C1B7	C1B8	C1B9	C1BA	C1BB	C1BC	C1BD	C1BE	C1BF	C1C0	C1C1	C1C2
졋	졍	졔	조	족	존	졸	졺	좀	좁	좃	종	좆	좋	좋	좌
C1C3	C1C4	C1C5	C1C6	C1C7	C1C8	C1C9	C1CA	C1CB	C1CC	C1CD	C1CE	C1CF	C1D0	C1D1	C1D2
좍	좔	좝	좟	좡	좨	좩	좽	죄	죈	죌	죔	죕	죗	죙	죠
C1D3	C1D4	C1D5	C1D6	C1D7	C1D8	C1D9	C1DA	C1DB	C1DC	C1DD	C1DE	C1DF	C1E0	C1E1	C1E2
죡	죤	죵	주	죽	준	줄	줅	줌	줍	줏	중	줒	쥐	쥑	쥔
C1E3	C1E4	C1E5	C1E6	C1E7	C1E8	C1E9	C1EA	C1EB	C1EC	C1ED	C1EE	C1EF	C1F0	C1F1	C1F2
쥘	쥠	쥡	쥣	쥥	쥧	쥩	즈	쥬	쥴	쥼	즈	즉	즌	즐	즘
C1F3	C1F4	C1F5	C1F6	C1F7	C1F8	C1F9	C1FA	C1FB	C1FC	C1FD	C1FE	C2A1	C2A2	C2A3	C2A4
즙	즛	즁	지	직	진	짇	질	짊	짐	집	짓	징	짖	짇	짚
C2A5	C2A6	C2A7	C2A8	C2A9	C2AA	C2AB	C2AC	C2AD	C2AE	C2AF	C2B0	C2B1	C2B2	C2B3	C2B4
짜	짝	짠	짢	짤	짧	짬	짭	짜	짝	짠	째	짹	짼	짤	짬
C2B5	C2B6	C2B7	C2B8	C2B9	C2BA	C2BB	C2BC	C2BD	C2BE	C2BF	C2C0	C2C1	C2C2	C2C3	C2C4
짭	짯	짱	짯	쨔	쨘	쨔	쩌	쩍	쩐	쩔	쩍	쩜	쩝	쩝	쩍
C2C5	C2C6	C2C7	C2C8	C2C9	C2CA	C2CB	C2CC	C2CD	C2CE	C2CF	C2D0	C2D1	C2D2	C2D3	C2D4
쩨	쩽	쩌	쩍	쪼	쪽	쪼	쭐	쪼	쭙	쪼	쪼	쪼	짜	짝	짤
C2D5	C2D6	C2D7	C2D8	C2D9	C2DA	C2DB	C2DC	C2DD	C2DE	C2DF	C2E0	C2E1	C2E2	C2E3	C2E4
짧	쨋	쨌	쩍	쩐	쩔	쩜	쩝	쩡	쭈	쭉	쭌	쭐	쭘	쭙	쭝
C2E5	C2E6	C2E7	C2E8	C2E9	C2EA	C2EB	C2EC	C2ED	C2EE	C2EF	C2F0	C2F1	C2F2	C2F3	C2F4
쥐	쥑	쥐	쥐	쯔	쯔	쯔	쯤	쯔	쯩	찌	찍	찐	찔	찜	찝	찡
C2F5	C2F6	C2F7	C2F8	C2F9	C2FA	C2FB	C2FC	C2FD	C2FE	C3A1	C3A2	C3A3	C3A4	C3A5	C3A6
찍	찢	차	착	찬	찮	찰	참	찹	찻	찼	창	찾	채	책	챈

Figure 84. Hangeul (7 of 10)

C3A7	C3A8	C3A9	C3AA	C3AB	C3AC	C3AD	C3AE	C3AF	C3B0	C3B1	C3B2	C3B3	C3B4	C3B5	C3B6
챌	챔	챕	챗	챘	챙	챠	챤	챦	챨	챰	챱	챳	처	척	천	철
C3B7	C3B8	C3B9	C3BA	C3BB	C3BC	C3BD	C3BE	C3BF	C3C0	C3C1	C3C2	C3C3	C3C4	C3C5	C3C6
첨	첩	첫	첬	청	체	첵	첸	첼	쳄	쳅	쳇	쳉	쳐	쳔	쳘
C3C7	C3C8	C3C9	C3CA	C3CB	C3CC	C3CD	C3CE	C3CF	C3D0	C3D1	C3D2	C3D3	C3D4	C3D5	C3D6
쳬	쳰	쳱	초	촉	촌	촐	촘	촙	촛	총	촤	촨	촬	촹	최
C3D7	C3D8	C3D9	C3DA	C3DB	C3DC	C3DD	C3DE	C3DF	C3E0	C3E1	C3E2	C3E3	C3E4	C3E5	C3E6
쵠	쵤	쵬	쵭	쵯	쵱	쵸	쵼	쵿	추	축	춘	출	춤	춥	춧	충
C3E7	C3E8	C3E9	C3EA	C3EB	C3EC	C3ED	C3EE	C3EF	C3F0	C3F1	C3F2	C3F3	C3F4	C3F5	C3F6
춰	췄	췌	췐	취	췬	췰	췸	췹	췻	췽	츄	츈	츌	츔	츕
C3F7	C3F8	C3F9	C3FA	C3FB	C3FC	C3FD	C3FE	C4A1	C4A2	C4A3	C4A4	C4A5	C4A6	C4A7	C4A8
츠	측	츤	츨	츰	츱	츳	층	치	칙	친	칟	칠	칡	침	칩
C4A9	C4AA	C4AB	C4AC	C4AD	C4AE	C4AF	C4B0	C4B1	C4B2	C4B3	C4B4	C4B5	C4B6	C4B7	C4B8
칫	칭	카	칵	칸	칼	캄	캅	캇	캉	캐	캑	캔	캘	캠	캡
C4B9	C4BA	C4BB	C4BC	C4BD	C4BE	C4BF	C4C0	C4C1	C4C2	C4C3	C4C4	C4C5	C4C6	C4C7	C4C8
캣	캤	캥	캬	캭	컁	커	컥	컨	컫	컬	컴	컵	컷	컸	컹
C4C9	C4CA	C4CB	C4CC	C4CD	C4CE	C4CF	C4D0	C4D1	C4D2	C4D3	C4D4	C4D5	C4D6	C4D7	C4D8
케	켁	켄	켈	켐	켑	켓	켕	켜	켠	켤	켬	켭	켯	켰	켱
C4D9	C4DA	C4DB	C4DC	C4DD	C4DE	C4DF	C4E0	C4E1	C4E2	C4E3	C4E4	C4E5	C4E6	C4E7	C4E8
켸	코	콕	콘	콜	콤	콥	콧	콩	콰	콱	콴	콸	콤	콩	쾌
C4E9	C4EA	C4EB	C4EC	C4ED	C4EE	C4EF	C4F0	C4F1	C4F2	C4F3	C4F4	C4F5	C4F6	C4F7	C4F8
쾡	쾨	쾰	쿄	쿠	쿡	쿤	쿨	쿰	쿱	쿳	쿵	쿼	퀀	퀄	퀑
C4F9	C4FA	C4FB	C4FC	C4FD	C4FE	C5A1	C5A2	C5A3	C5A4	C5A5	C5A6	C5A7	C5A8	C5A9	C5AA
퀘	퀭	퀴	퀵	퀸	퀼	퀴	퀍	퀴	퀴	큐	큐	큘	큠	큐	큭
C5AB	C5AC	C5AD	C5AE	C5AF	C5B0	C5B1	C5B2	C5B3	C5B4	C5B5	C5B6	C5B7	C5B8	C5B9	C5BA
큰	큘	큼	큽	큿	키	킥	킨	킬	킴	킵	킷	킹	타	탁	탄
C5BB	C5BC	C5BD	C5BE	C5BF	C5C0	C5C1	C5C2	C5C3	C5C4	C5C5	C5C6	C5C7	C5C8	C5C9	C5CA
탈	탉	탐	탑	탓	탔	탕	태	택	탠	탤	탬	탭	탯	탰	탱
C5CB	C5CC	C5CD	C5CE	C5CF	C5D0	C5D1	C5D2	C5D3	C5D4	C5D5	C5D6	C5D7	C5D8	C5D9	C5DA
탸	탼	터	턱	턴	털	턺	텀	텁	텃	텄	텅	테	텍	텐	텔
C5DB	C5DC	C5DD	C5DE	C5DF	C5E0	C5E1	C5E2	C5E3	C5E4	C5E5	C5E6	C5E7	C5E8	C5E9	C5EA
템	텝	텟	텡	텨	텬	텼	톄	톈	토	톡	톤	톨	톰	톱	톳

Figure 85. Hangeul (8 of 10)

C5EB	C5EC	C5ED	C5EE	C5EF	C5F0	C5F1	C5F2	C5F3	C5F4	C5F5	C5F6	C5F7	C5F8	C5F9	C5FA
뚱	뚜	뚸	뚼	뚸	뚸	뚼	뚼	뚼	뚜	뚜	뚜	뚜	뚤	뚬	뚭
C5FB	C5FC	C5FD	C5FE	C6A1	C6A2	C6A3	C6A4	C6A5	C6A6	C6A7	C6A8	C6A9	C6AA	C6AB	C6AC
뚯	뚱	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚜	뚜	뚤	뚬
C6AD	C6AE	C6AF	C6B0	C6B1	C6B2	C6B3	C6B4	C6B5	C6B6	C6B7	C6B8	C6B9	C6BA	C6BB	C6BC
뚭	뚜	뚤	뚤	뚤	뚤	뚤	뚤	뚤	뚤	뚸	뚼	뚼	뚼	뚼	뚸
C6BD	C6BE	C6BF	C6C0	C6C1	C6C2	C6C3	C6C4	C6C5	C6C6	C6C7	C6C8	C6C9	C6CA	C6CB	C6CC
뚸	뚼	뚼	뚼	뚼	뚼	뚼	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C6CD	C6CE	C6CF	C6D0	C6D1	C6D2	C6D3	C6D4	C6D5	C6D6	C6D7	C6D8	C6D9	C6DA	C6DB	C6DC
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C6DD	C6DE	C6DF	C6E0	C6E1	C6E2	C6E3	C6E4	C6E5	C6E6	C6E7	C6E8	C6E9	C6EA	C6EB	C6EC
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C6ED	C6EE	C6EF	C6F0	C6F1	C6F2	C6F3	C6F4	C6F5	C6F6	C6F7	C6F8	C6F9	C6FA	C6FB	C6FC
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C6FD	C6FE	C7A1	C7A2	C7A3	C7A4	C7A5	C7A6	C7A7	C7A8	C7A9	C7AA	C7AB	C7AC	C7AD	C7AE
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C7AF	C7B0	C7B1	C7B2	C7B3	C7B4	C7B5	C7B6	C7B7	C7B8	C7B9	C7BA	C7BB	C7BC	C7BD	C7BE
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C7BF	C7C0	C7C1	C7C2	C7C3	C7C4	C7C5	C7C6	C7C7	C7C8	C7C9	C7CA	C7CB	C7CC	C7CD	C7CE
뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸	뚸
C7CF	C7D0	C7D1	C7D2	C7D3	C7D4	C7D5	C7D6	C7D7	C7D8	C7D9	C7DA	C7DB	C7DC	C7DD	C7DE
하	한	한	할	할	함	합	하	하	해	해	해	해	해	해	해
C7DF	C7E0	C7E1	C7E2	C7E3	C7E4	C7E5	C7E6	C7E7	C7E8	C7E9	C7EA	C7EB	C7EC	C7ED	C7EE
해	해	햐	향	허	헌	헌	헌	헌	헌	헌	헌	헌	헤	헤	헤
C7EF	C7F0	C7F1	C7F2	C7F3	C7F4	C7F5	C7F6	C7F7	C7F8	C7F9	C7FA	C7FB	C7FC	C7FD	C7FE
헤	헤	헤	헤	헤	혀	현	현	현	현	현	현	현	현	혜	혜
C8A1	C8A2	C8A3	C8A4	C8A5	C8A6	C8A7	C8A8	C8A9	C8AA	C8AB	C8AC	C8AD	C8AE	C8AF	C8B0
혜	혜	호	홍	홍	홀	홍	홈	홉	홍	홍	홑	화	확	환	활
C8B1	C8B2	C8B3	C8B4	C8B5	C8B6	C8B7	C8B8	C8B9	C8BA	C8BB	C8BC	C8BD	C8BE	C8BF	C8C0
활	황	홰	홰	홰	홰	홰	회	획	획	획	획	획	획	횡	효
C8C1	C8C2	C8C3	C8C4	C8C5	C8C6	C8C7	C8C8	C8C9	C8CA	C8CB	C8CC	C8CD	C8CE	C8CF	C8D0
효	효	효	후	후	후	훌	훌	훔	후	후	훠	훤	훨	훰	훵

Figure 86. Hangeul (9 of 10)

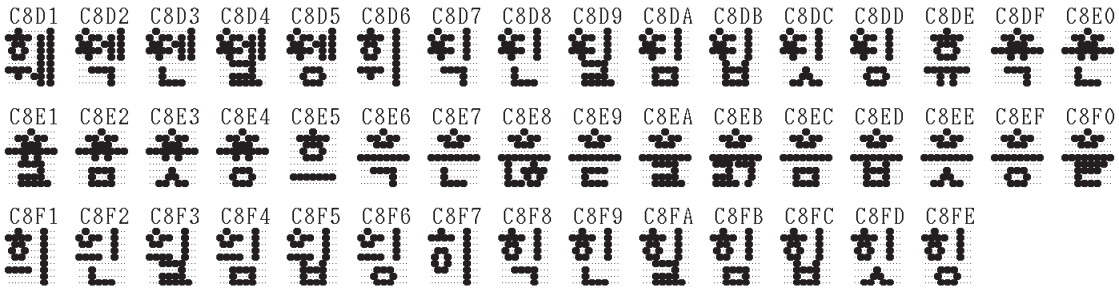


Figure 87. Hangeul (10 of 10)

Code Page 1088

HEX DIGITS 1ST → 2ND ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		┼ SF050000	(SP) SP010000	0 ND100000	@ SM050000	P LP020000	` SD130000	p LP010000								
-1	┌ SF010000	◄ SM630000	! SP020000	1 ND010000	A LA020000	Q LQ020000	a LA010000	q LQ010000								
-2	┐ SF030000	↕ SM760000	" SP040000	2 ND020000	B LB020000	R LR020000	b LB010000	r LR010000								
-3	└ SF020000	!! SP330000	# SM010000	3 ND030000	C LC020000	S LS020000	c LC010000	s LS010000								
-4	┘ SF040000	¶ SM250000	$ SC030000	4 ND040000	D LD020000	T LT020000	d LD010000	t LT010000								
-5	│ SF110000	┴ SF070000	% SM020000	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000								
-6	─ SF100000	┬ SF060000	& SM030000	6 ND060000	F LF020000	V LV020000	f LF010000	v LV010000								
-7	• SM570000	┤ SF090000	' SP050000	7 ND070000	G LG020000	W LW020000	g LG010000	w LW010000								
-8	◘ SM570001	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000								
-9	○ SM750000	├ SF080000	) SP070000	9 ND090000	I LI020000	Y LY020000	i LI010000	y LY010000								
-A	◙ SM750002	→ SM310000	* SM040000	: SP130000	J LJ020000	Z LZ020000	j LJ010000	z LZ010000								
-B	♂ SM280000	← SM300000	+ SA010000	; SP140000	K LK020000	[SM060000	k LK010000	{ SM110000								
-C	♀ SM290000	∟ SA420000	, SP080000	< SA030000	L LL020000	₩ SC140000	l LL010000	| SM130000								
-D	♪ SM930000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000								
-E	♫ SM910000	▲ SM600000	. SP110000	> SA050000	N LN020000	^ SD150000	n LN010000	~ SD190000								
-F	☼ SM690000	▼ SV040000	/ SP120000	? SP150000	O LO020000	_ SP090000	o LO010000	⌂ SM790000								

Code Page 01088

Appendix A. JavaPOS support for UnifiedPOS device statistics properties

Common properties

Table 232. Description of common properties

Common Properties	Description	Value
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	UPOS Version (for example: 1.9.1)
DeviceCategory	Device category (for example: POSPrinter)	[Corresponding category]
ManufacturerName	Device manufacturer's name	IBM
ModelName	Device model name	See the section for the specific device.
SerialNumber	Device serial number	[Serial_Number]
ManufactureDate	Device manufacture date	See the section for the specific device.
FirmwareRevision	Device firmware revision	[Firmware_Version]
Interface	Device hardware interface (for example: USB)	RS232/RS485/USB/Proprietary
InstallationDate	Device installation date	Not supported
HoursPoweredCount	Number of hours powered On	Not supported
CommunicationErrorCount	Number of communication errors	Not supported

Notes:

1. Interface property used the *Proprietary* value for PS/2 and Embedded devices.
2. SerialNumber property for USB devices may not match the device label, except for the 4610 POSPrinter.
3. The ManufactureDate format for the [Printer/MICR/CheckScanner/CashDrawer] is *WWYY*, where *WW* is a two-digit representation for the week of the year the printer was manufactured, and *YY* is the last two digits of the year it was manufactured.
4. EIA232 hardware interface use RS232 as a value.

Support for the following properties applies to all devices:

- CapStatisticsReporting: Yes
- CapUpdateStatistics: No

Cash drawer

Table 233. Cash drawer

System/device	Bus	Model name value	
		Port 3A	Port 3B
SurePOS 730/750	USB	4800-73X/75X - A	4800-73X/75X - B
POSPrinter 4610	USB / EIA232 / RS485	4610 - A	4610 - B
SurePOS 300	Embedded	4800-72x/74x/78x, 4810-3xx	4800-72x/74x/78x, 4810-3xx
SurePOS 500/600/Kiosk	EIA232	4840-xx1/xx1/xx3 - A	4840-xx1/xx1/xx3 - A
SurePOS 72x/74x/78x	Embedded	4800-72x/74x/78x, 4810-3xx	4800-72x/74x/78x, 4810-3xx
SurePOS 4694	RS485	4694-2xx/3xx - A	4694-2xx/3xx - B
SurePOS 4674	Embedded	4674 - A	4674 - B
SureOne 4614/4615 SurePOS 100	Embedded	461x	
AnyPlace POS Hub	USB	AnyPlace Hub	

Table 234. Cash drawer common properties

Common Properties	SurePOS 730/750 USB	POS Printer 4610 EIA232/ RS485/ USB	SurePOS 300 Embed-ded	SurePOS 500/ 600/ Kiosk EIA232	SurePOS 72x/ 74x/ 78x Embed-ded	SurePOS 4694 - RS485	SurePOS 4674 Embed-ded	SureOne/ SurePOS 100 Embed-ded	AnyPlace POS Hub - USB
UnifiedPOSVersion	Y	Y	Y	Y	Y	Y	Y	Y	Y
DeviceCategory	Y	Y	Y	Y	Y	Y	Y	Y	Y
ManufacturerName	Y	Y	Y	Y	Y	Y	Y	Y	Y
ModelName	Y	Y	Y	Y	Y	Y	Y	Y	Y
SerialNumber	Y	Y	N	N	N	N	N	N	Y
ManufactureDate	N	N	N	N	N	N	N	N	N
Mechanical Revision	N	N	N	N	N	N	N	N	N
FirmwareRevision	Y	Y	N	N	N	N	N	N	N
Interface	Y	Y	Y	Y	Y	Y	Y	Y	Y
InstallationDate	N	N	N	N	N	N	N	N	N
HoursPowered Count	N	N	N	N	N	N	N	N	N
Communication ErrorCount	N	N	N	N	N	N	N	N	N

Table 235. Cash drawer specific properties

Specific Properties	SurePOS 730/750 USB	POS Printer 4610 EIA232/ RS485/ USB	SurePOS 300 Embed-ded	SurePOS 500/ 600/ Kiosk EIA232	SurePOS 72x/ 74x/ 78x Embed-ded	SurePOS 4694 - RS485	SurePOS 4674 Embed-ded	SureOne/ SurePOS 100 Embe-ded	Any-Place POS Hub - USB
DrawerGoodOpen Count	N	N	N	N	N	N	N	N	N
DrawerFailedOpen Count	N	N	N	N	N	N	N	N	N

Note: 730/750 USB devices use BCD level for Firmware Revision.

Check scanner

Check Scanner is only available on IBM 4610 PosPrinter models T18 and T19.

Table 236. Check scanner

Device	Bus	Model name value
POSPrinter 4610-TI8/TI9	USB / EIA232 / RS485	4610

Table 237. Check scanner common properties

Common Properties	Description	POSPrinter 4610 -TI8/TI9 - EIA232/RS485/USB Supported:
UnifiedPOSVersion	Version of UnifiedPOS specification supported	Y
DeviceCategory	Device Category	Y
ManufacturerName	Device manufacturer's name	Y
ModelName	Device model name	Y
SerialNumber	Device serial number	Y
ManufactureDate (*WWYY* format)	Device manufacture date	Y
MechanicalRevision	Device hardware revision	N
FirmwareRevision	Device firmware revision	Y
Interface	Device hardware interface	Y
InstallationDate	Device installation date	N
HoursPoweredOn	Number of hours powered On	N
CommunicationErrorCount	Number of communication errors	N

Table 238. Check scanner manufacturer properties

Manufacturer properties	Unit of measure	Description	POSPrinter 4610 -TI8/TI9 - EIA232/RS485/USB supported:
IBM_CheckScannedCount	”” - empty	Number of checks scanned	Y

Table 238. Check scanner manufacturer properties (continued)

Manufacturer properties	Unit of measure	Description	POSPrinter 4610 -TI8/TI9 - EIA232/RS485/USB supported:
IBM_ChecksFailedQualityCount	″″ - empty	Number of scanned checks failed to meet internal image quality test	Y
IBM_CheckScannerBrightnessQuality	″″ - empty	Check Scanner Brightness Quality	Y
IBM_CheckScannerContrastQuality	″″ - empty	Check Scanner Contrast Quality	Y
IBM_CheckScannerFocusQuality	″″ - empty	Check Scanner Focus Quality	Y

Notes:

1. CheckScanner microcode level used for firmware revision
2. ManufacturerDate is only available for new POSPrinters (after December, 2005). Older ones are not supported.
3. The CheckScanner should be cleaned or recalibrated if the following values are not met:
   - IBM_CheckScannerBrightnessQuality: 90 decimal
   - IBM_CheckScannerContrastQuality: 90 decimal
   - IBM_CheckScannerFocusQuality: 77 decimal
4. For 4689 USB and RS485 models, PaperCutCount increments its value by 10 after 10 cuts, not one by one.

Fiscal printer

Table 239. Fiscal printer

Device	Bus	Model name value
4610-KCx	EIA232	KCX*
4610-KDx	EIA232	KDX*
4610-GDx	EIA232	GDX*
3FA	RS485	3Fx/GRX/KRX/KSX
3F2	RS485	3Fx/GRX/KRX/KSX
3F	RS485	3Fx/GRX/KRX/KSX
3FB	RS485	3Fx/GRX/KRX/KSX
3Bx	RS485	3Fx/GRX/KRX/KSX
GRx	RS485	3Fx/GRX/KRX/KSX
KRx	RS485	3Fx/GRX/KRX/KSX
KSx	RS485	3Fx/GRX/KRX/KSX
4610-GEx	USB	GEX/GBX/GH5
4610-GBx	USB	GEX/GBX/GH5
4610-KH5	USB	KH5
4610-GH5	USB	GEX/GBX/GH5

Table 240. Fiscal printer common properties

Common Properties	USB	EIA232	RS485
UnifiedPOSVersion	Y	Y	Y
DeviceCategory	Y	Y	Y
ManufacturerName	Y	Y	Y
ModelName	Y	Y	Y
SerialNumber	Y	N	N
ManufactureDate	N	N	N
MechanicalRevision	N	N	N
FirmwareRevision	Y	N	N
Interface	Y	Y	Y
InstallationDate	N	N	N
HoursPoweredOn	N	N	N
CommunicationErrorCount	N	N	N

Table 241. Fiscal printer specific properties

Specific Properties	USB	EIA232	RS485
BarcodePrintedCount	N	N	N
FormInsertionCount	N	N	N
HomeErrorCount	N	N	N
JournalCharacterPrinted Count	N	N	N
JournalLinePrintedCount	N	N	N
MaximumTempReached Count	N	N	N
NVRAMWriteCount	N	N	N
PaperCutCount	N	N	N
FailedPaperCutCount	N	N	N
PrinterFaultCount	N	N	N
PrintSideChangeCount	N	N	N
FailedPrintSideChange Count	N	N	N
ReceiptCharacterPrinted Count	N	N	N
ReceiptCoverOpenCount	N	N	N
ReceiptLineFeedCount	N	N	N
ReceiptLinePrintedCount	N	N	N
SlipCharacterPrintedCount	N	N	N
SlipCoverOpenCount	N	N	N
SlipLineFeedCount	N	N	N
SlipLinePrintedCount	N	N	N
StampFiredCount	N	N	N

Note: USB devices use BCD level for firmware revision.

Hard totals

Table 242. Hard Totals

Device	Bus	Model name value
NVRAM 730/750	USB	4800-73X/75X
NVRAM 72X/74X/78X	Embedded	4800-72x/74x/78x
4694	Embedded	4694
4674	Embedded	4674

Table 243. Hard totals common properties

Common properties	Description	Support USB	Support embedded
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y
DeviceCategory	Device category	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y
ModelName	Device model name	Y	Y
SerialNumber	Device serial number	Y	N
ManufactureDate	Device manufacture date	N	N
MechanicalRevision	Device hardware revision	N	N
FirmwareRevision	Device firmware revision	Y	N
Interface	Device hardware interface	Y	Y
InstallationDate	Device installation date	N	N
HoursPoweredCount	Number of hours powered On	N	N
CommunicationErrorCount	Number of communication errors	N	N

Note: USB devices use BCD level for firmware revision.

Keylock

Table 244. Keylock

Device	Bus	Model name value
NANPOS Keylock	USB/RS485	NANPOS
NANPOS Keylock as System attached	USB/PS2	NANPOS
4820 Keylock (SurePoint)	USB/RS485/PS2	4820
50-key Keylock SBCS/DBCS	USB/RS485	50-key
ANKPOS - DBCS	USB/RS485	ANKPOS
ANKPOS - DBCS as System Keyboard	USB/PS2	ANKPOS
133-key or Matrix - SBCS	USB/RS485	133-key

Table 244. Keylock (continued)

Device	Bus	Model name value
Keyboard 4685 K02 Ultra VI	USB/RS485	4685
Keyboard 4685 K02 Ultra VI with MSR/E	RS485	4685
Keyboard V	USB/RS485	Kbd-V
SurePOS 4674 Keylock	RS485	4674
SureOne 4614 / 4615 Keylock SurePOS 100 – Keylock	Embedded	461x
Keyboard 4685-K03	USB	4685-K03
Keyboard 4685-K03	RS-485	4674/4685-K03
ANPOS Keylock	PS/2, USB	Modular ANPOS
Modular CANPOS Keylock	PS/2, USB	Modular CANPOS
Modular 67 Key Keylock	PS/2, USB	Modular 67 Key

Table 245. Keylock common properties

Common properties	Description	USB supported	RS485 supported	PS2 supported	Embedded
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y	Y
ModelName	Device model name	Y	Y	Y	Y
SerialNumber	Device serial number	Y	N	N	N
ManufactureDate	Device manufacture date. In WWYY format.	Y[1]	N	N	N
MechanicalRevision	Device hardware revision	N	N	N	N
FirmwareRevision	Device firmware revision	Y	N	N	N
Interface	Device hardware interface	Y	Y	Y	Y
InstallationDate	Device installation date	N	N	N	N
HoursPoweredCount	Number of hours powered On	Y[1]	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Table 246. Keylock manufacturer properties table

Manufacturer properties	Description	USB	RS485	PS2	Embedded
IBM_PowerCycleCount	Number of power on/off cycles	Y[1]	N	N	N
IBM_UnexpectedResetsCount	Number of unexpected resets	Y[1]	N	N	N

[1] Supported by USB Modular ANPOS, 67 Key, Modular CANPOS

Table 247. Keylock specific properties

Specific properties	Description	USB supported	RS485 supported	PS2 supported	Embedded
LockPositionChangeCount	Number of lock position changes	Y[1]	N	N	N

[1] Supported by USB Modular ANPOS, 67 Key, Modular CANPOS

Note: USB devices use BCD level for firmware revision.

Line display

Table 248. Line display

Device	Bus	Model name value
2X20 VFD 1-sided	USB/EIA232/RS485	2X20 VFD
2X20 VFD 2-sided	USB/RS485	2X20 VFD - A or 2X20 VFD - B
2X20 LCD	USB/RS485	2X20 LCD
APA Display	USB/EIA232/RS485	APA
50-key Keyboard Display LCD	USB/RS485	2X20 LCD on 50-key
PLU Keyboard III with APA Display	USB/RS485	APA

Table 249. Line Display common properties

Common properties	Description	USB supported	EIA232 supported	RS485 supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y
ModelName	Device model name	Y	Y	Y
SerialNumber	Device serial number	Y	N	N
ManufactureDate	Device manufacture date	N	N	N
MechanicalRevision	Device hardware revision	N	N	N
FirmwareRevision	Device firmware revision	Y	N	N
Interface	Device hardware interface	Y	Y	Y
InstallationDate	Device installation date	N	N	N
HoursPoweredCount	Number of hours powered On	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N

Table 250. Line Display specific properties

Specific properties	Description	USB supported	EIA232 supported	RS485 supported
OnlineTransitionCount	Number of online transitions (on after screen blanking)	N	N	N

Note: USB devices use BCD level for firmware revision.

MICR

Table 251. MICR

Device	Bus	Model name value
POSPrinter 4610-TI4/5/8/9/2CR	EIA232/RS485/USB	4610

Table 252. MICR common properties

Common properties	Description	EIA232/RS485/USB supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y
DeviceCategory	Device category	Y
ManufacturerName	Device manufacturer's name	Y
ModelName	Device model name	Y
SerialNumber	Device serial number	Y
ManufactureDate (*WWYY* format)	Device manufacture date	Y
MechanicalRevision	Device hardware revision	N
FirmwareRevision	Device firmware revision	Y
Interface	Device hardware interface	Y
InstallationDate	Device installation date	N
HoursPoweredCount	Number of hours powered On	N
CommunicationErrorCount	Number of communication errors	N

Table 253. MICR specific properties

Specific properties	Description	EIA232/RS485/USB supported
GoodReadCount	Number of successful reads	Y
FailedReadCount	Number of failed reads	Y
FailedDateParseCount	Number of failed data parses	N

Table 254. MICR manufacturer properties

Manufacturer Properties	Description	EIA232/RS485/USB supported
IBM_CurrentMICRNoiseValue	Value of the MICR Noise on the last read operation	Y (Only 2CR model)

Notes:

1. POSPrinter microcode level used for Firmware Revision.
2. ManufacturerDate is only available for newer POSPrinters (after December 2005). Older ones are not supported.
3. DeviceCategory value is taken from the programmatic name.

Motion sensor

Table 255. Motion sensor

Device	Bus	Model name value
SurePOS 500/600/AnyPlace Kiosk	Embedded	IBM Generic Motion Sensor

Table 256. Motion sensor common properties

Common Properties	Description	Embedded Supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y
DeviceCategory	Device category	Y
ManufacturerName	Device manufacturer's name	Y
ModelName	Device model name	Y
SerialNumber	Device serial number	N
ManufactureDate	Device manufacture date	N
MechanicalRevision	Device hardware revision	N
FirmwareRevision	Device firmware revision	N
Interface	Device hardware interface	Y
InstallationDate	Device installation date	N
HoursPoweredCount	Number of hours powerd On	N
CommunicationErrorCount	Number of communication errors	N

Table 257. Motion Sensor specific properties

Specific properties	Description	Supported
MotionEventCount	Number of motion occurrences	N

MSR

Table 258. MSR

Device	Bus	Model name value
ANPOS ISO MSR	USB	NANPOS - ISO
ANPOS as System attached ISO MSR	PS2/USB	NANPOS - ISO
4820 ISO MSR (SurePoint)	USB	4820 - ISO
4820 JUCC MSR (SurePoint)	USB	4820 - JUCC
4820/AnyPlace Kiosk/SurePOS 500/600 ISO MSR	EIA232	4820/500/600/AnyPlace Kiosk – ISO
4820/AnyPlace Kiosk/SurePOS 500/600 JUCC MSR	EIA232	4820/500/600/AnyPlace Kiosk – JUCC
50-key ISO MSR	USB	50-key - ISO
CANPOS ISO MSR	PS2	CANPOS - ISO
ANKPOS JUCC MSR	USB	ANKPOS - JUCC

Table 258. MSR (continued)

Device	Bus	Model name value
ANKPOS as System keyboard JUCC MSR	PS2/USB	ANKPOS - JUCC
133-key or Matrix - ISO MSR	USB	133-key ISO
Keyboard 4685 K02 Ultra VII - JUCC MSR 4 pos	RS485/USB	4685/Kbd V JUCC
Keyboard 4685 K02 Ultra VII - JUCC MSR/E 4 pos	RS485	4685 JUCC with Encoder
Keyboard 4685 K02 Ultra VII - JUCC MSR/E 6 pos	RS485	4685 JUCC with Encoder
Keyboard V JUCC MSR	RS485/USB	4685/Kbd V - JUCC
133-key/4820/50-key/NANPOS ISO MSR	RS485	IBM MSR - ISO
SurePOS 4674/4820/50-key/ANKPOS JUCC MSR	RS485	IBM MSR - JUCC
SureOne 4614/4615 ISO MSR SurePOS 100 ISO MSR	PS2	461x - ISO
50-key JUCC MSR	USB	50-key - JUCC
Keyboard 4685 K03 JUCC MSR	USB	4685-K03 - JUCC
Keyboard 4685 K03 JUCC MSR	RS485	4685 / Kbd V - JUCC
Modular ANPOS with ISO MSR	PS/2, USB	Modular ANPOS-ISO
Modular ANPOS with JUCC MSR	PS/2, USB	Modular ANPOS-JUCC
Modular CANPOS with ISO MSR	PS/2, USB	Modular CANPOS-ISO
Modular CANPOS with JUCC MSR	PS/2, USB	Modular CANPOS-JUCC
Modular 67 Key with ISO MSR	PS/2, USB	Modular 67 Key-ISO
Modular 67 Key with JUCC MSR	PS/2, USB	Modular 67 Key-JUCC

Table 259. MSR common properties

Common properties	Description	USB supported	PS2 supported	EIA232 supported	RS485 supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y	Y
ModelName	Device model name	Y	Y	Y	Y
SerialNumber	Device serial number	Y	N	N	N
ManufactureDate	Device manufacture date. In WWYY format.	Y[1]	N	N	N
MechanicalRevision	Device hardware revision	N	N	N	N
FirmwareRevision	Device firmware revision	Y	N	N	N
Interface	Device hardware interface	Y	Y	Y	Y
InstallationDate	Device installation date	N	N	N	N
HoursPoweredCount	Number of hours powered On	Y[1]	N	N	N

Table 259. MSR common properties (continued)

Common properties	Description	USB supported	PS2 supported	EIA232 supported	RS485 supported
CommunicationErrorCount	Number of communication errors	N	N	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Table 260. MSR manufacturer properties table

Manufacturer properties	Description	USB	RS485	EIA232	RS485
IBM_PowerCycleCount	Number of power on/off cycles	Y[1]	N	N	N
IBM_UnexpectedResetsCount	Number of unexpected resets	Y[1]	N	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Table 261. MSR specific properties

Specific properties	Description	USB supported	PS2 supported	EIA232 supported	RS485 supported
GoodReadCount	Number of successful reads	Y[1]	N	N	N
FailedReadCount	Number of failed reads	Y[1]	N	N	N
UnreadableCardCount	Number of failed data parses	Y[1]	N	N	N
MissingStartSentinelTrack1Count	Number of Reads that Miss Start Stentinel	Y[1]	N	N	N
ParityLRCErrorTrack1Count	Number of LRC Error	Y[1]	N	N	N
MissingStartSentinelTrack2Count	Number of Reads that Miss Start Stentinel	Y[1]	N	N	N
ParityLRCErrorTrack2Count	Number of LRC Error	Y[1]	N	N	N
MissingStartSentinelTrack3Count	Number of Reads that Miss Start Stentinel	Y[1]	N	N	N
ParityLRCErrorTrack3Count	Number of LRC Error	Y[1]	N	N	N
MissingStartSentinelTrack4Count	Number of Reads that Miss Start Stentinel	Y[1]	N	N	N

Table 261. MSR specific properties (continued)

Specific properties	Description	USB supported	PS2 supported	EIA232 supported	RS485 supported
ParityLRCErrorTrack4Count	Number of LRC Error	Y[1]	N	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Note: USB devices use BCD level for firmware revision.

POS keyboard

Table 262. POS keyboard

Device	Bus	Model name value
NANPOS POSKeyboard - SBCS	RS485/USB	NANPOS
NANPOS POSKeyboard - System attached	PS2/USB	NANPOS
4820 POSKeyboard (SurePoint-keypad)	PS2/RS485/USB	Keypad
50-key POSKeyboard - SBCS	RS485/USB	50-key
CANPOS POSKeyboard - SBCS	PS2	CANPOS
ANKPOS - DBCS	RS485/USB	ANKPOS
ANKPOS - DBCS System Keyboard	PS2/USB	ANKPOS
133-key or Matrix - SBCS	RS485/USB	133-key
PLU Keyboard III with Display	RS485/USB	PLU
Keyboard 4685 K02 Ultra VII - 4 pos Keylock	RS485/USB	4685
Keyboard 4685 K02 Ultra VII - MSR/E 4 pos Keylock	RS485	4685
Keyboard 4685 K02 Ultra VII - MSR/E 6 pos Keylock	RS485	4685
Keyboard V	RS485/USB	Kbd V
SurePOS 4674 POSKeyboard	RS485	4674/4685-K03
SureOne 4614/4615 Keyboard SurePOS 100 – Keyboard	PS2	461x
50-key POSKeyboards JUCC MSR - DBCS	RS485/USB	50-key
Keyboard 4685 K03	USB	4685-K03
Keyboard 4685 K03	RS485	4674/4685-K03
Modular ANPOS Keyboard	PS/2, USB	Modular ANPOS
Modular CANPOS Keyboard	PS/2, USB	Modular CANPOS
Modular 67 Key Keyboard	PS/2, USB	Modular 67 Key

Table 263. POS Keyboard common properties

Common properties	Description	USB supported	PS2 supported	RS485 supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y
ModelName	Device model name	Y	Y	Y
SerialNumber	Device serial number	Y	N	N
ManufactureDate	Device manufacture date. In WWYY format	Y[1]	N	N
MechanicalRevision	Device hardware revision	N	N	N
FirmwareRevision	Device firmware revision	Y	N	N
Interface	Device hardware interface	Y	Y	Y
InstallationDate	Device installation date	N	N	N
HoursPoweredCount	Number of hours powered On	Y[1]	N	N
CommunicationErrorCount	Number of communication errors	N	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Table 264. POS keyboard manufacturer properties table

Manufacturer properties	Description	USB	PS2	RS485
IBM_PowerCycleCount	Number of power on/off cycles	Y[1]	N	N
IBM_UnexpectedResetsCount	Number of unexpected resets	Y[1]	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Table 265. POS Keyboard specific properties

Specific properties	Description	USB supported	PS2 supported	RS485 supported
KeyPressedCount	Number of keys pressed	Y[1]	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards

Note: USB devices use BCD level for firmware revision.

POS printer

Table 266. POS printer

Device	Bus	Model name value
4610 TI3 (thermal/impact)	EIA232/RS485/USB	4610-TI3/4
Device: 4610 2NR (thermal/impact) - SBCS	EIA232/RS485/USB	4610-2NR-SBCS

Table 266. POS printer (continued)

Device	Bus	Model name value
4610 2CR (2NR+flipper/MICR) - SBCS	EIA232/RS485/USB	4610-2CR-SBCS
4610 2NR (Thermal/impact) - DBCS	EIA232/RS485/USB	4610-2NR-DBCS
4610 2CR (2NR+flipper/MICR) - DBCS	EIA232/RS485/USB	4610-2CR-DBCS
4610 TI4 (TI3+flipper/MICR)	EIA232/RS485/USB	4610-TI3/4
4610 TI5 (DBCD TI3)	EIA232/RS485/USB	4610-TI5
4610 TI8	EIA232/RS485/USB	4610-TI8
4610 TI9	EIA232/RS485/USB	4610-TI9
4610 TI1/2 (thermal/impact)	EIA232/RS485	4610-TI1/2
4610 Tx6 (SS thermal)	EIA232/RS485/USB	4610-Tx6
4610 Tx7 (SS thermal DBCS)	EIA232/RS485/USB	4610-Tx7
4689 TI5 all except 001 and 002 (DBCS)	RS485/USB	4689
4674 Embedded Printer - same as 4689	RS485	4674
SureOne 4614/4615 Thermal SurePOS 100 – Thermal	EIA232	461x-Thermal
SureOne 4613/4614/4615 Impact SurePOS 100 – Impact	EIA232	461x-Impact

Table 267. POS printer common properties

Common properties	Description	4610 Printer USB/EIA232/ RS485 supported	4674/4689 RS485	4689 USB	SureONE Impact /Thermal
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y	Y
ModelName	Device model name	Y	Y	Y	Y
SerialNumber	Device serial number	Y	N	Y	N
ManufactureDate (*WWYY* format)	Device manufacture date	Y	N	N	N
MechanicalRevision	Device hardware revision	N	N	N	N
FirmwareRevision	Device firmware revision	Y	Y	Y	N
Interface	Device hardware interface	Y	Y	Y	Y
InstallationDate	Device installation date	N	N	N	N
HoursPoweredCount	Number of hours powered On	N	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

Table 268. POS printer specific properties

Specific properties	Description	4610 Printer USB/EIA232/ RS485 supported	4674/4689 RS485	4689 USB	SureONE Thermal
BarCodePrintedCount	Number of barcodes printed	Y	N	N	N
FormInsertionCount	Number of forms inserted into the document/slip station	Y	N	N	N
HomeErrorCount	Number of home errors	Y	N	N	N
JournalCharacterPrintedCount	Number of journal characters printed	N	N	N	N
JournalLinePrintedCount	Number of journal lines printed	N	N	Y	N
MaximumTempReachedCount	Number of times maximum temperature reached	Y	N	N	N
NVRAMWriteCount	Number or times NVRAM is written to	Y	N	N	N
PaperCutCount	Number of paper cuts	Y	Y	Y	N
FailedPaperCutCount	Number of failed paper cuts	Y (See note 6.)	N	N	N
PrinterFaultCount	Number of printer faults	N	N	N	N
PrintSideChangeCount	Number of print side changes (or check flips) performed	Y (See note 5.)	N	N	N
FailedPrintSideChangeCount	Number of print side change (of check flip) failures	Y (See note 5.)	N	N	N
ReceiptCharacterPrinted Count	Number of receipt characters printed	Y	N	N	N
ReceiptCoverOpenCount	Number of times receipt cover was opened	Y	N	N	N
ReceiptLineFeedCount	Number of receipt line feeds performed	Y	N	N	N
ReceiptLinePrintedCount	Number of receipt lines printed	N	Y	Y	N
SlipCharacterPrintedCount	Number of document/slip characters printed	Y	N	N	N
SlipCoverOpenCount	Number of times the document/slip station cover opened	Y	N	N	N
SlipLineFeedCount	Number of document/slip line feeds performed	Y	N	N	N
SlipLinePrintedCount	Number of document/slip lines printed	N	N	N	N
StampFiredCount	Number of stamps fired	N	N	N	N

Notes:

1. POSPrinter microcode level used for firmware revision.

2. ManufacturerDate is only available for newer POSPrinters (after December 2005). Older ones are not supported.
3. For 4689 USB and RS485 models, PaperCutCount increments by 10 after 10 cuts.
4. In some cases a tag <Value/> may be retrieved, indicating that the requested statistic is not supported for the printer firmware, or it has never been initialized.
5. Only valid on 4610-TI4/8/9 models.
6. Only valid on Tx6/Tx7/2xR models.

Table 269. POS printer manufacturer properties

Manufacturer Properties		**RS232/RS485/ USB**
Name	Description	Supported
IBM_ReceiptPaperRemaining	Number of mm of paper left in the printer	Yes, see note[1]
IBM_UnexpectedSlipCoverOpenCount	Number of times the Slip cover was opened without an error condition.	Yes, see note[1]
IBM_SlipFeedErrorCount	Number of Slip Feed errors	Yes, see note[1]
IBM_ReceiptPaperJamCount	Number of Receipt Paper Jams	Yes, see note[1]
IBM_FlashWriteFailedCount		Yes, see note[1]
IBM_UnexpectedRecCoverOpenCount	Number of failed flash writes	Yes, see note[1]
IBM_ImpactHeadFailedCount	Number of times the Receipt cover was opened without an error condition.	Yes, see note[1]
IBM_ThermalPrintHeadElementFailure	Number of Impact head failed coils Number of the Thermal Head element that is failing (Can be more than 1).	Yes, see note[1]

[1]Only valid on 4610-2xR models

Scale

Table 270. Scale

Device	**Bus**	**Model name value**	**Manufacturer**
OEM PSC Magellan Scanner/Scale Model 384	USB	OEM - Scale	OEM
OEM PSC Magellan Scanner/Scale Model 8201	USB	OEM - Scale	OEM
IBM 4696 Model 1	RS485	4696	OEM
IBM 4698 Model 2	RS485	4698	OEM

Table 271. Scale common properties

Common properties	**Description**	**USB supported**	**RS485 supported**
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y
DeviceCategory	Device category	Y	Y

Table 271. Scale common properties (continued)

Common properties	Description	USB supported	RS485 supported
ManufacturerName	Device manufacturer's name	Y	Y
ModelName	Device model name	Y	Y
SerialNumber	Device serial number	Y	N
ManufactureDate	Device manufacture date	N	N
MechanicalRevision	Device hardware revision	N	N
FirmwareRevision	Device firmware revision	Y	N
Interface	Device hardware interface	Y	Y
InstallationDate	Device installation date	N	N
HoursPoweredCount	Number of hours powered On	N	N
CommunicationErrorCount	Number of communication errors	N	N

Table 272. Scale specific properties

Specific properties	Description	USB supported	RS485 supported
GoodWeightReadCount	Number of successful weight reads	N	N

Note: USB devices use BCD level for Firmware Revision.

Scanner

Table 273. Scanner

Device	Bus	Model name value	Manufacturer
OEM PSC Magellan Scanner/Scale - Model 384	USB	OEM-Scanner	OEM
OEM PSC Magellan Scanner/Scale - Model 8201	USB	OEM-Scanner	OEM
OEM Symbol Hand Held - Model LS4804	USB	OEM-Scanner	OEM
OEM Symbol Hand Held - Model LS2104	USB	OEM-Scanner	OEM
OEM Symbol Hand Held - Model M2004	RS485	OEM-Scanner	OEM
IBM Hand Held BCR Model 1 (4500)	RS485	4500/4501	OEM
IBM Hand Held BCR Model 2 (4501)	RS485	4500/4501	OEM
IBM 1520 Hand Held Model A02	RS485	1520	OEM
IBM 4685 Hand Held Model 001/K001	RS485	4685	OEM
IBM 4696 Scanner/Scale Model 1	RS485	4696	OEM
IBM 4697 Model 1	RS485	4697	OEM
IBM 4698 Model 1/2	RS485	4698	OEM
AnyPlace Kiosk - 4838 Line	EIA232	Line	OEM

Table 273. Scanner (continued)

Device	Bus	Model name value	Manufacturer
AnyPlace Kiosk - Omni	EIA232	Omni	OEM

Table 274. Scanner common properties

Common properties	Description	USB supported	RS485 supported	EIA232
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y
ModelName	Device model name	Y	Y	Y
SerialNumber	Device serial number	Y	N	N
ManufactureDate	Device manufacture date	N	N	N
MechanicalRevision	Device hardware revision	N	N	N
FirmwareRevision	Device firmware revision	Y	N	N
Interface	Device hardware interface	Y	Y	Y
InstallationDate	Device installation date	N	N	N
HoursPoweredCount	Number of hours powered On	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N

Table 275. Scanner specific properties

Specific properties	Description	USB supported	RS485 supported	EIA232
GoodScanCount	Number of successful scans	N	N	N

Notes:

1. USB devices use BCD level for firmware revision.
2. DeviceCategory value is taken from programmatic name.

Tone indicator

Table 276. Tone indicator

Device	Bus	Model name value
NANPOS Tone Indicator - SBCS	RS485/USB	NANPOS
NANPOS Tone Indicator as System Keyboard	PS2/USB	NANPOS
4820 Tone Indicator (SurePoint)	PS2/RS485/USB	4820
50-key Tone Indicator - SBCS	RS485/USB	50-key
ANKPOS - DBCS	RS485/USB	ANKPOS
ANKPOS - DBCS as System Keyboard	PS2/USB	ANKPOS
133-key or Matrix - SBCS	RS485/USB	133-key
PLU Keyboard III with Display	RS485/USB	PLU

Table 276. Tone indicator (continued)

Device	Bus	Model name value
Keyboard 4685 K02 Ultra VII - 4 pos Keylock	RS485/USB	4685
Keyboard 4685 K02 Ultra VII - MSR/E 4 pos	USB	4685
Keyboard 4685 K02 Ultra VII - MSR/E 6 pos	USB	4685
Keyboard V	RS485/USB	Kbd V
SurePOS 4674 Tone Indicator	RS485	4674/4685-K03
50-key POSKeyboard JUCC MSR - DBCS	USB	50-key
4610 Tx6/Tx7 Printer ToneIndicator	EIA232/RS485/USB	4610-Tx6/Tx7
Keyboard 4685 K03	USB	4685-K03
Keyboard 4685 K03	RS485	4674/4685-K03
Modular ANPOS Keyboard	PS/2, USB	Modular ANPOS
Modular CANPOS Keyboard	PS/2, USB	Modular CANPOS
Modular 67 Key Keyboard	PS/2, USB	Modular 67 Key

Table 277. Tone indicator common properties

Common properties	Description	4610 Tx6/Tx7 Printer tone indicator	USB supported	PS2 supported	RS485 supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y	Y	Y	Y
DeviceCategory	Device category	Y	Y	Y	Y
ManufacturerName	Device manufacturer's name	Y	Y	Y	Y
ModelName	Device model name	Y	Y	Y	Y
SerialNumber	Device serial number	Y	Y	N	N
ManufactureDate	Device manufacture date. In WWYY format.	Y[1]	N	N	N
MechanicalRevision	Device hardware revision	N	N	N	N
FirmwareRevision	Device firmware revision	Y	Y	N	N
Interface	Device hardware interface	Y	Y	Y	Y
InstallationDate	Device installation date	N	N	N	N
HoursPoweredCount	Number of hours powered On	Y[1]	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

[1] Supported by USB Modular ANPOS, 67 Key, Modular CANPOS

Table 278. Tone Indicator manufacturer properties table

Manufacturer properties	Description	4610 Tx6/Tx7 Printer ToneIndicator	USB	PS2	Embedded
IBM_PowerCycleCount	Number of power on/off cycles	N	Y[1]	N	N
IBM_UnexpectedResetsCount	Number of unexpected resets	N	Y[1]	N	N

[1] Supported by USB Modular ANPOS, Modular CANPOS and Modular 67 Key Keyboards.

Table 279. Tone indicator specific properties

Specific properties	Description	4610 Tx6/Tx7 Printer tone indicator	USB supported	PS2 supported	RS485 supported
ToneSoundedCount	Number of tones played	N	N	N	N

Note: USB devices use BCD level for firmware revision.

Appendix B. UnifiedPOS support tool

To start the application from a command prompt, type:

```
java com.ibm.jpos.tools.sdist.SupportToolApp
```

In Windows, you can click the **UnifiedPOS Support Tool** icon under **Start > Programs > IBM JavaPOS**.

At startup, after the splash screen, the first dialog appears as shown:

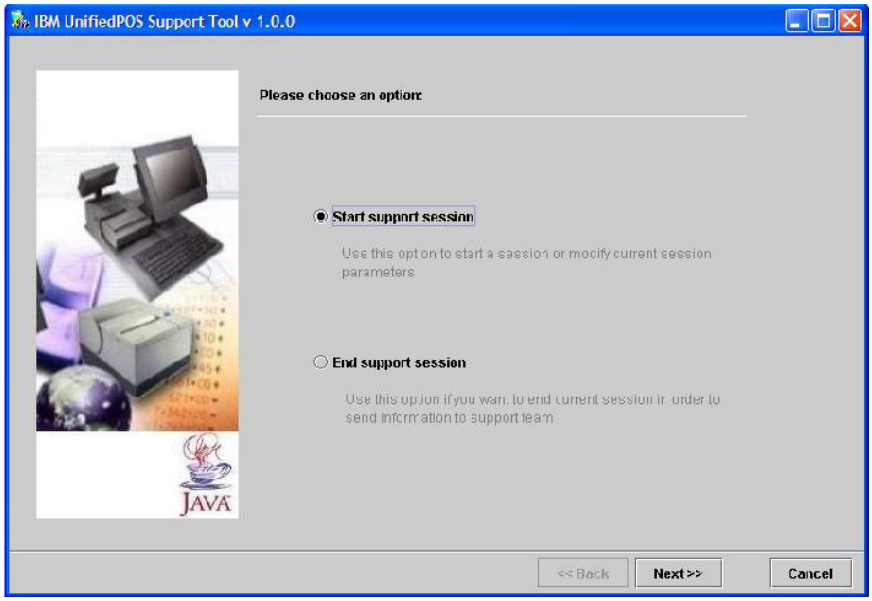


Figure 88. UnifiedPOS Support Tool GUI

Start support session is used to start a new session or to edit current session settings. *End support session* is used to finalize the current support session, automatically disable traces and generate the session files that are sent to the support team.

When you choose *Start support session*, the following window is presented:

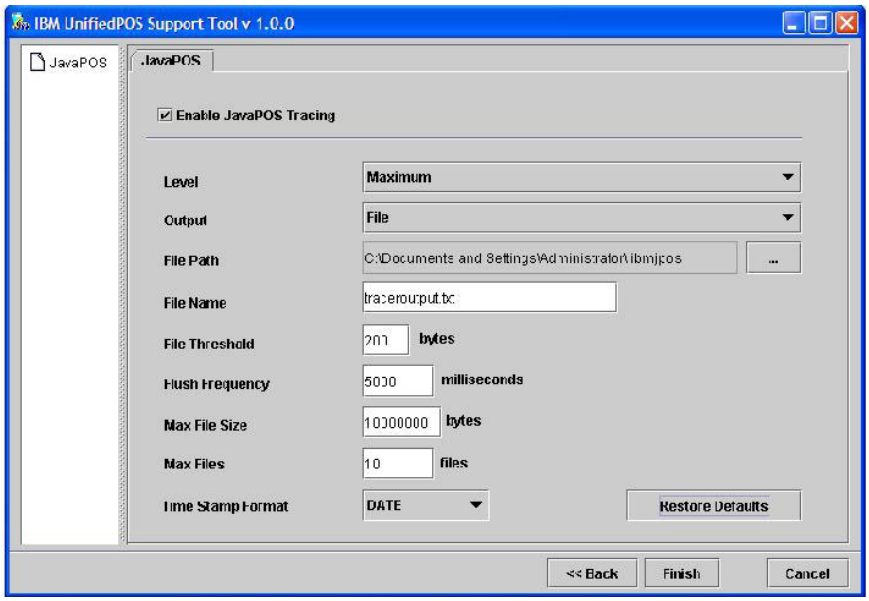


Figure 89. Session options

By default, *Enable JavaPOS Tracing* is checked to enable tracing. If you want disable tracing, uncheck this field. *Restore Defaults* loads all the default factory values in one step. Click **Finish** to save your changes. The *End Support Session* option generates the session files and displays a success/fail message, and the location of the generated file.

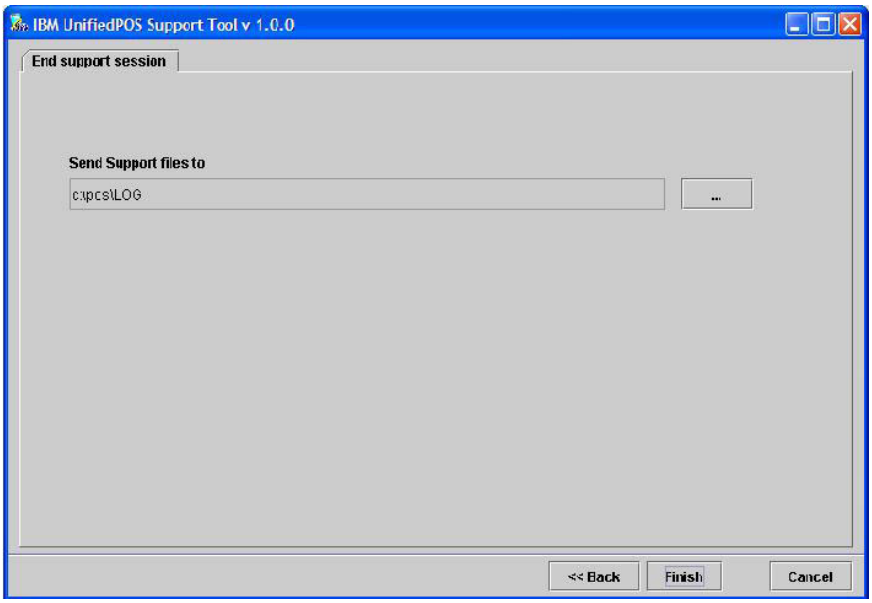


Figure 90. End session

When you press the <...> (explore button) the following dialog is presented, and only directories can be selected:

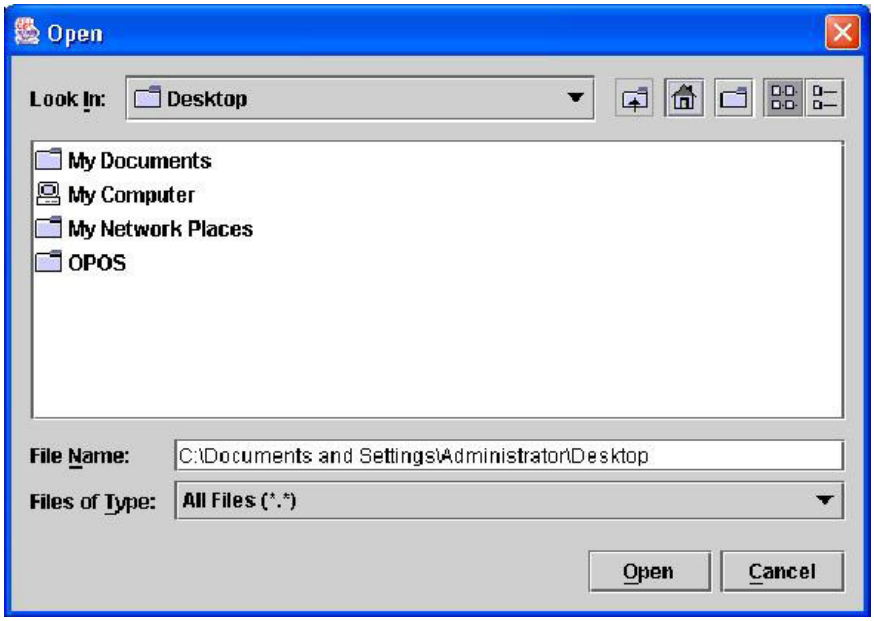


Figure 91. Environment variables file name

After clicking **Finish**, a file named: `send`*`YYYYMMDDHHMMSS`*`.zip` is generated.

Where:

- *YYYYMMDD* = year, month and day when the file was generated.
- *HHMMSS* = hour, minute and second when the file was generated. This zip file contains all the information relative to environment variables, jpos.xml, libraries, and tracer output.

Appendix C. IBM Java Web Start configuration to access the POS control center

Install JavaPOS SDK

Create a KeyStore

To create a store key:

1. Change to webstart directory

   For Windows: **cd c:\POS\IBMJPOS\webstart**

   Note: There is assumed the product is installed in the default directory. Linux: cd /opt/ibm/javapos/webstart

   For Linux: **cd /opt/ibm/javapos/webstart**

   Note: This and following steps is assumed they are executed with root user.
2. For Windows:

   Execute keytool.exe -genkey -keystore jposKeyStore -alias jposAlias

   For Linux:

   Execute keytool -genkey -keystore jposKeyStore -alias jposAlias

Expected input:

```
Enter keystore password:  "keyStorePassword"

What is your first and last name?
[Unknown]:  "userName"

What is the name of your organizational unit?
[Unknown]:  "yourOrganizationalUnit"

What is the name of your organization?
[Unknown]:  "yourOrganization"

What is the name of your City or Locality?
[Unknown]:  "yourCity"

What is the name of your State or Province?
[Unknown]:  "yourState"

What is the two-letter country code for this unit?
[Unknown]:  "us"

Is CN=userName, OU=yourOrganizationUnit, O=yourOrganization,
L=yourCity, ST=yourState, C=us correct?
type "yes" or "no") [no]:  yes

Enter key password
for <jposAlias>: RETURN if same as keystore password):"aliasPassword"
```

Create a self-signed certificate

To create a create a self-signed certificate:

1. Execute this command, **keytool.exe -list -v -keystore jposKeyStore .**
2. Enter keystore password: `keyStorePassword`

Expected input:

```
Keystore type: jks
Keystore provider: IBMJCE

Your keystore contains 1 entry

Alias name: jposAlias
Creation date: Apr 9, 2008
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=userName, OU=yourOrganizationUnit,
O=yourOrganization, L=yourCity, ST=yourState, C=us
Issuer: CN=userName, OU= yourOrganizationUnit,
O= yourOrganization, L= yourCity, ST= yourState, C=us
Serial number: 47fcfb8e
Valid from: 4/9/08 2:23 PM until: 7/8/08 2:23 PM
Certificate fingerprints:
  MD5:  BB:A4:36:45:C4:0D:FD:A0:0D:66:12:4F:D0:B3:D8:EE
  SHA1:
  BE:D4:D4:7A:50:A6:A3:8B:F9:D6:38:2E:0E:BD:15:D1:9B:83:B7:24
```

Create jpos.xml

Create jpos.xml and save in the following directory:

For Windows: **cd c:\POS\IBMJPOS\config**

For Linux: **cd /opt/ibm/javapos/etc**

Create the resource JAR

To create the resource JAR:

1. Change directory
   Windows: **cd c:\POS\IBMJPOS\config**
   Linux: **cd /opt/ibm/javapos/etc**
2. Execute the appropriate command:
   Windows: **jar -cvf c:\POS\IBMJPOS\webstart\ibmjpos_resources.jar *.***
   Linux: **jar -cvf /opt/ibm/javapos/webstart/ibmjpos_resources.jar *.***
3. Change directory (only for Linux)
   Windows
   No action is necessary
   Linux **cd /opt/javax-usb/etc**
4. Execute the appropriate command (Only for Linux)
   Windows
   No action is necessary
   Linux **jar -uf /opt/ibm/javapos/webstart/ibmjpos_resources.jar**

Sign the all JARs file with the test certificate

To sign the all JARs file with the test certificate

1. Copy all the jars located in:
   Windows: `C:\POS\IBMJPOS\lib` to`C:\POS\IBMJPOS\webstart` directory.
   Linux: `/opt/ibm/javapos/lib` and `/opt/javax-usb/lib` to `/opt/ibm/javapos/webstart` directory.

Note: It is necessary to copy all the jar files for the JavaPOS device drivers to the webstart directory. Otherwise, once the existing jar files are signed, a java application that is not launched through the webstart interface will not be able to use the JavaPOS device drivers.

2. Sign all jars located at webstart directory.

   For example, to sign the jtux.jar use the following command: **jarsigner.exe -keystore jposKeyStore –storepass “keyStorePassword” -keypass “aliasPassword” jtux.jar jposAlias**

Run JNLP file

To run the JNLP file

1. Move to:

   Windows: `cd  c:\POS\IBMJPOS\webstart`

   Linux: `cd /opt/ibm/javapos/webstart`

2. Run.

   For Windows or Linux: **Javaws ibmJavaPOS.jnlp**.

   Note: When UnifiedPOS was installed without ″System Management Support″, edit ibmJavaPOS.jnlp and comment out the following line: `<!jar href="jpos_sysmgmt.jar" download="eagerly" />`

Notices

This information was developed for products and services offered in the U.S.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering the subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statementmay not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Trademarks

The following are trademarks of the IBM Corporation in the United States or other countries, or both:

IBM	PS/2
the IBM logo	OS/2®
SureMark	SurePoint™
SurePOS™	SureOne®

Microsoft, Windows, Windows NT® are trademarks of Microsoft Corporation in the United States and/or other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary includes terms and definitions from:

- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

A

active. Able to communicate on the network. A token-ring network adapter is active if it is able to transmit and receive on the network. Operational. Pertaining to a node, or device that is connected or is available for connection to another node or device. Currently transmitting or receiving.

adapter. In the point-of-sale terminal, a circuit card that, with its associated software, enables the terminal to use a function or feature. In a LAN, within a communicating device, a circuit card that, with its associated software and/or microcode, enables the device to communicate over the network.

address. In data communication, the IEEE-assigned unique code, or the unique locally administered code assigned to each device, or workstation connected to a network. A character, group of characters, or a value that identifies a register, a particular part of storage, a data source, or a data link. The value is represented by one or more characters. To refer to a device, or an item of data by its address. The location in the storage of a computer where data is stored.

address space. The complete range of addresses that is available to a programmer.

all points addressable (APA). In computer graphics, pertaining to the ability to address and display or not display each picture element (pel) on a display surface.

alphanumeric. Pertaining to a character set containing letters, digits, and other characters, such as punctuation marks.

Alphanumeric point-of-sale keyboard (NANPOS keyboard). This keyboard consists of a section of alphanumeric keys, a programmable set of point-of-sale keys, a numeric keypad, and system function keys.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphics characters.

American National Standards Institute (ANSI). An organization for the purpose of establishing voluntary industry standards.

NANPOS keyboard. Alphanumeric Point of Sale Keyboard.

ANSI. American National Standards Institute.

APA. all points addressable.

API. Application program interface.

application program. A program written for or by a user that applies to the user's own work. A program written for or by a user that applies to a particular application. A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface (API). The formally defined programming language interface that is between an IBM system control program or a licensed program and the user of the program.

array. An arrangement of elements in one or more dimensions.

ASCII. American National Standard Code for Information Interchange.

asynchronous. Pertaining to two or more processes that do not depend upon the occurrence of specific events such as timing signals. Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

attach. To connect a device physically. To make a device a part of a network logically. Compare with *connect*.

attaching device. Any device that is physically connected to a network and can communicate over the network.

B

backup. Pertaining to a system, device, file, or facility that can be used in the event of a malfunction or the loss of data.

backup copy. A copy, usually of a program or of a library member, that is kept in case the original or the working copy is unintentionally altered or destroyed.

bar code. A code representing characters by sets of parallel bars of varying thickness and separation that are read optically by transverse scanning.

BCD. Binary-coded decimal notation.

binary. Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. Pertaining to a selection, choice, or condition that has two possible different values or states.

binary-coded decimal notation (BCD). A binary-coded notation in which each of the decimal digits is represented by a binary numeral. For example, in binary-coded decimal notation that uses the weights 8, 4, 2, 1, the number “twenty three” is represented by 0010 0011. In the pure binary numeration system, its representation is 10111.

bit. Either of the binary digits: a 0 or 1.

bit map. A coded representation in which each bit or group of bits represents or corresponds to an item; for example, a configuration of bits in main storage in which each bit indicates whether a peripheral device or a storage block is available or in which each group of bits corresponds to one pixel of a display image.

bits per second (bps). The rate at which bits are transmitted per second.

block size. The minimum size that frames are grouped into for retransmission. The number of data elements (such as bits, bytes, characters, or records) that are recorded or transmitted as a unit.

break scan code. The hardware scan code received by the keyboard device driver when a key on the keyboard is physically pressed.

bps. Bits per second.

Bps. Bytes per second.

buffer. A portion of storage used to hold input or output data temporarily. A routine or storage used to compensate for a difference in data rate or time of occurrence of events, when transferring data from one device to another.

byte. A string that consists of a number of bits, treated as a unit, and representing a character. A binary character operated upon as a unit and usually shorter than a computer word. A string that consists of a particular number of bits, usually 8, that is treated as a unit, and that represents a character. A group of 8 adjacent binary digits that represent one extended binary-coded decimal interchange code (EBCDIC). See *n-bit byte*.

C

C. A high-level programming language designed to optimize run time, size, and efficiency.

call. The action of bringing a function or subprogram into effect, usually by specifying the entry conditions and jumping to an entry point.

card reader. See *magnetic stripe reader, (MSR)*.

cash drawer. A drawer at a point-of-sale terminal that can be programmed to open automatically. See *till*.

channel. A functional unit, controlled by a host computer, that handles the transfer of data between processor storage and local peripheral equipment. A path along which signals can be sent. The portion of a storage medium that is accessible to a given reading or writing station.

clear. To delete data from a screen or from memory.

code page. A particular assignment of hexadecimal identifiers to graphic characters.

code point. A 1-byte code representing one of 256 potential characters.

command. A request for performance of an operation or execution of a program. A character string from a source external to a system that represents a request for system action.

compile. To translate all or part of a program expressed in a high-level language into a computer program expressed in an intermediate language, an assembly language, or a machine language. To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one computer instruction for each symbolic statement, or both, as well as performing the function of an assembler. To translate a source program into an executable program (an object program). To

translate a program written in a high-level programming language into a machine language program.

compiler. A program that decodes instructions written as pseudo codes and produces a machine language program to be executed at a later time. Contrast with *interpretive routine*.

component. Any part of a network other than an attaching device, such as an IBM 8228 Multistation Access Unit. Hardware or software that is part of a functional unit.

configuration. The group of devices, options, and programs that make up a data processing system or network as defined by the nature, number, and chief characteristics of its functional units. More specifically, the term refer to a hardware configuration or a software configuration. See also *system configuration*.

configuration file. The collective set of definitions that describes a configuration.

connect. In a LAN, to physically join a cable from a station to an access unit or network connection point. Contrast with *attach*.

constant. String or numeric value that does not change throughout program execution.

control character. A character whose occurrence in a particular context initiates, modifies, or stops a control operation. A control character may be recorded for use in a subsequent action, and it may have a graphic representation in some circumstances.

CRC. Cyclic redundancy check.

customize. To tailor a program or store system through option selection.

cyclic redundancy check (CRC). Synonym for *frame check sequence (FCS)*.

D

data. A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means. Any representations such as characters or analog quantities to which meaning is or might be assigned.

data communication. Transfer of information between functional units by means of data transmission according to a protocol. The transmission, reception, and validation of data.

data file. A collection of related data records organized in a specific manner; for example, a payroll file (one record for each employee, showing such information as rate of pay and deductions) or an inventory file (one record for each inventory item, showing such information as cost, selling price, and number in stock.) See also *data set, file*.

data set. Logically related records treated as a single unit. See also *file*.

data terminal equipment (DTE). That part of a data station that serves as a data source, data receiver, or both. Equipment that sends or receives data, or both.

data transmission. The conveying of data from one place for reception elsewhere by means of telecommunications.

data type. The mathematical properties and internal representation of data and functions.

DBCS. Double-byte character set.

DCE. Data circuit-terminating equipment.

default. Pertaining to an attribute, value, or option that is assumed when none is explicitly specified.

default value. The value the system supplies when the user does not specify a value.

device. A mechanical, electrical, or electronic contrivance with a specific purpose. An input/output unit such as a terminal, display, or printer. See also *attaching device*.

device connection. The connection between an application and a hardware device created by the IBM JavaPOS system when the application opens a device.

device descriptor. An identifier that represents a device to the IBM JavaPOS system application programming interface. This identifier is created by the IBM JavaPOS system when the application opens a device.

device driver. The code needed to attach and use a device on a computer or a network.

digital. Pertaining to data in the form of digits. Contrast with *analog*. Pertaining to data consisting of numerical values or discrete units.

direct file. A file in which records are assigned specific record positions. No matter what order the records are put in a direct file, they always occupy the assigned position. A direct file is the same as a random file except that a direct file contains no delimiting characters, such as quotes enclosing string fields.

directory. A table of identifiers and references that correspond to items of data. An index that a control program uses to locate one or more blocks of data that are stored in separate areas of a data set in direct access storage.

disabled. Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line.

disk. A round, flat plate coated with a magnetic substance that is used to store computer data. See also *integrated disk, fixed disk*.

Disk Operating System (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

display. A visual presentation of data. A device that presents visual information to the point-of-sale terminal operator and to the customer, or to the display station operator.

distributed. Physically separate but connected by cables.

DLL. See *dynamic link library*.

DOS. Disk Operating System.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with single-byte character set.

DRAM. Dynamic RAM. See *RAM*.

driver. Software component that controls a device.

DTE. Data terminal equipment.

dump. To write at a particular instant the contents of storage, or part of storage, onto another data medium for the purpose of safeguarding or debugging the data. Data that has been dumped.

duplex. In data communication, pertaining to a simultaneous two-way independent transmission in both directions. Synonymous with *full-duplex*. contrast with *half-duplex*.

DVD-ROM. Digital-video-disk read-only memory.

dynamic link library (DLL). In the Windows operating systems, the delayed connection of a library to a routine until load time or run time.

E

EAN. European article number.

EIA. Electronics Industries Association.

EIA-232. In data communications, a specification of the Electronic Industries Association (EIA) that defines the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) using serial binary data interchange. Formerly known as RS-232.

enabled. On a LAN, pertaining to an adapter or device that is active, operational, and able to receive frames from the network. Pertaining to a state of a processing unit that allows the occurrence of certain types of interruptions. Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line.

end-of-file. An internal label, immediately following the last record of a file, signaling the end of that file.

error message. A message that is issued because an error has been detected.

escape character. Code extension character used, in some cases, with one or more succeeding characters to indicate by some convention that the coded representation following the character or the group of characters are to be interpreted according to a different code or different character set.

European article number (EAN). A number that is assigned to and encoded on an article of merchandise for scanning in some countries.

event. Processing unit containing price changes and item file updates. All records in an event share common characteristics such as type of change and event due date. An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an I/O operation.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file. See also *overflow exception* and *underflow exception*.

exit. To execute an instruction or statement within a portion of a program in order to terminate the execution of that portion. **Note:** Such portions of programs include loops, routines, subroutines, and modules.

expansion board. In an IBM Personal Computer, a panel containing microchips that a user can install in an expansion slot to add memory or special features. Synonymous with *expansion card*, *extender card*.

expansion card. Synonym for *expansion board*.

extender card. Synonym for *expansion board*.

F

fat-finger. When two keys are pressed faster than the value specified using the PosNfatFingerTimeOut resource. This could occur under any of the following conditions: 1) Two keys on the keyboard were pressed

at the same time. 2) The operator is keying faster than 25 keys per second. 3) A double key is not defined to the keyboard device handler.

field. On a data medium or a storage medium, a specified area used for a particular category of data; for example, a group of character positions used to enter or display wage rates on a panel.

FIFO. First-in–first-out.

file. A named set of records stored or processed as a unit. For example, an invoice form a record and the complete set of such records form a file. See also *data file* and *data set*.

file name. A name assigned or declared for a file. The name used by a program to identify a file.

first-in–first-out (FIFO). A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time.

fixed disk (drive). In a personal computer system unit, a disk storage device that reads and writes on rigid magnetic disks. It is faster and has a larger storage capacity than a diskette and is permanently installed.

flag. A character or indicator that signals the occurrence of some condition, such as the setting of a switch, or the end of a word.

flash memory. A data-storage device that is programmable, erasable, and does not require continuous power. The chief benefit of flash memory over other programmable and erasable data storage devices is that it can be reprogrammed without being removed from the circuit board.

frame. The unit of transmission in some LANs, including the IBM Token-Ring Network and the IBM PC Network. It includes delimiters, control characters, information, and checking characters. On a token-ring network, a frame is created from a token when the token has data appended to it. On a token-bus network (IBM PC Network), all frames including the token frame contain a preamble, start delimiter, control address, optional data and checking characters, end delimiter, and are followed by a minimum silence period. A housing for machine elements. In synchronous data link control (SDLC), the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. Each frame begins and ends with a flag.

frequency. The rate of signal oscillation, expressed in hertz (cycles per second).

full-duplex. Synonym for *duplex*.

function. A specific purpose of an entity, or its characteristic action. A subroutine that returns the value of a single variable. In data communications, a machine action such as a carriage return or line feed.

G

GCGID. See *Graphic Character Global Identifier*.

global. Pertaining to that which is defined in one subdivision of a computer program and used in at least one other subdivision of that computer program.

Graphic Character Set Global Identifier (GCGID). A 4- to 8-character identifier assigned to a registered graphic character in an IBM registry.

group. A set of related records that have the same value for a particular field in all records. A collection of users who can share access authorities for protected resources. A list of names that are known together by a single name.

H

half-duplex. In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*.

hardware. Physical equipment as opposed to programs, procedures, rules, and associated documentation.

hertz (Hz). A unit of frequency equal to one cycle per second. **Note:** In the United States, line frequency is 60Hz or a change in voltage polarity 120 times per second; in Europe, line frequency is 50Hz or a change in voltage polarity 100 times per second.

hexadecimal notation. Notation for the base-16 number system using the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to represent values from 0 to 15 (decimal).

hot key. The key combination used to change from one session to another on a workstation.

hot plug. To connect a USB I/O device to the universal serial bus without powering the host system down.

hot unplug. To disconnect a USB I/O device from the universal serial bus without powering the host system down.

Hz. See *hertz*.

I

IBM Disk Operating System (DOS). A disk operating system based on MS-DOS.[1]

identifier. String of characters used to name elements of a program, such as variable names, reserved words, and user-defined function names.

inactive. Not operational. Pertaining to a node or device not connected or not available for connection to another node or device. In the IBM Token-Ring Network, pertaining to a station that is only repeating frames or tokens, or both.

information (I) frame. A frame in I format used for numbered information transfer. See also *supervisory frame, unnumbered frame.*

initialize. In a LAN, to prepare the adapter (and adapter support code, if used) for use by an application program.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation.

input device. Synonym for *input unit*.

input/output device. See *I/O device*.

input/output (I/O). Pertaining to a device whose parts can perform an input process and an output process at the same time. Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

input unit. A device in a data processing system that is used to enter data into the system. Synonymous with *input device*.

instance. An occurrence of a particular device or object. For example, two instances of the `PosDisplay` class name can be `Shopper1` and `Shopper2`, where these instances refer to the same physical display.

integrated. Arranged together as one unit.

integrated disk. An integral part of the processor that is used for magnetically storing files, application programs, and diagnostics. Synonymous with *disk*.

interaction. A basic unit used to record system activity, consisting of the acceptance of a line of terminal input, processing of the line, and a response, if any.

interface. A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, and other characteristics as appropriate. A shared boundary. An interface may be a hardware component to link two devices or a portion of storage or registers accessed by two or more computer programs. Hardware, software, or both, that links systems, programs, or devices.

interleave. To insert segments of one program into another program so that the two programs can, in effect, be executed at the same time.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

interpretive routine. A routine that decodes instructions written as pseudocodes and immediately executes the instructions. Contrast with *compile*.

I/O. Input/output.

I/O device. Equipment for entering and receiving data from the system.

IPL. Initial program load.

IRES. IBM Retail Environment for SuSE Linux.

ISO. International Organization for Standardization.

item. One member of a group. In a store, one unit of a commodity, such as one box, one bag, or one can. Usually an item is the smallest unit of a commodity to be sold.

J

JIS. Japanese Industrial Standard

JUCC. Japanese Unified Cash Card

K

keyboard. A group of numeric keys, alphabetic keys, special character keys, or function keys used for entering information into the terminal and into the system.

kHz. Kilohertz. See also *hertz*.

kilohertz (kHz). A thousand hertz. See also *hertz*.

L

LED. Light-emitting diode.

lift-off. When a pointing device is removed from a touch-sensitive surface.

1. MS-DOS is a trademark of the Microsoft Corporation.

light-emitting diode (LED). A semiconductor chip that gives off visible or infrared light when activated.

line. On a terminal, one or more characters entered before a return to the first printing or display position.

link . The combination of physical media, protocols, and programming that connects devices on a network. In computer programming, the part of a program, in some cases a single instruction or an address, that passes control and parameters between separate portions of the computer program. To interconnect items of data or portions of one or more computer programs.

listing. A printout, usually prepared by a language translator, that lists the source code.

load. In computer programming, to enter data into memory or working registers.

lock. To disable a device, such as a scanner or MSR, so that it cannot receive input. See also *unlock*.

logging. The chronological recording of events occurring in a system or a subsystem for accounting or data collection purposes.

logical connection. In a network, devices that can communicate or work with one another because they share the same protocol. See also *physical connection*.

logon (n). The procedure for starting up a point-of-sale terminal or store controller for normal sales operations by sequentially entering the correct security number and transaction number. Synonymous with *sign-on*.

log on (v). To initiate a session. In SNA products, to initiate a session between an application program and a logical unit (LU). Synonymous with *sign-on*.

loop. A set of instructions that may be executed repeatedly while a certain condition prevails. See also *store loop*. A closed unidirectional signal path connecting input/output devices to a network.

M

macro. An instruction that causes the execution of a predefined sequence of instructions in the same source language.

magnetic stripe. The magnetic material (similar to recording tape) on merchandise tickets, credit cards, and employee badges. Information is recorded on the stripe for later “reading” by the magnetic stripe reader (MSR) or magnetic wand reader attached to the point-of-sale terminal.

magnetic stripe reader (MSR). A device that reads coded information from a magnetic stripe on a card, such as a credit card, as it passes through a slot in the reader.

make scan code. The hardware scan code received by the keyboard device driver when a key on the keyboard is physically pressed.

Mb. Megabit.

MB. Megabyte.

megabit (Mb). A unit of measure for throughput. 1 megabit = 1,048,576 bits.

megabyte (MB). A unit of measure for data. 1 megabyte = 1,048,576 bytes.

memory. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.

message. An arbitrary amount of information whose beginning and end are defined or implied. A group of characters and control bit sequences transferred as an entity. In telecommunication, a combination of characters and symbols transmitted from one point to another. A logical partition of the user device’s data stream to and from the adapter. See also *error message, operator message*.

microcode. One or more microinstructions. A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable. To design, write, and also test one or more microinstructions.

microprocessor. An integrated circuit that accepts coded instructions for execution. The instructions may be entered, integrated, or stored internally.

migration. Installation of a new version of a release of a program to replace an earlier version or release.

modem (Modulator/DEModulator). A device that converts digital data from a computer to an analog signal that can be transmitted in a telecommunication line, and converts the analog signal received to data for the computer.

modulo check. A function designed to detect most common input errors by performing a calculation on values entered into a system by an operator or scanning device.

monitor. A functional unit that observes and records selected activities for analysis within a data processing system. Possible uses are to show significant departures from the norm, or to determine levels of utilization of particular functional units. Software or hardware that observes, supervises, controls, or verifies operations of a system.

MSR. Magnetic stripe reader.

N

name. An alphanumeric term that identifies a data set, statement, program, or cataloged procedure.

n-bit byte. A string that consists of n bits.

network. A configuration of data processing devices and software connected for information interchange. An arrangement of nodes and connecting branches. Connections are made between data stations.

noise. A disturbance that affects a signal and that can distort the information carried by the signal. Random variations of one or more characteristics of any entity, such as voltage, current, or data. Loosely, any disturbance tending to interfere with normal operation of a device or system.

nonvolatile random access memory (NVRAM). Random access memory that retains its contents after electrical power is shut off.

NVRAM. nonvolatile random access memory.

O

offline. Operation of a functional unit without the control of a computer or control unit.

online. Operation of a functional unit that is under the continual control of a computer or control unit. The term also describes a user's access to a computer using a terminal.

open. To make an adapter ready for use. A break in an electrical circuit. To make a file ready for use.

operating system. Software that controls the execution of programs. An operating system provides services such as resource allocation, scheduling, input/output control, and data management.

operation. A defined action, namely, the act of obtaining a result from one or more operands in accordance with a rule that completely specifies the result for any permissible combination of operands. A program step undertaken or executed by a computer. An action performed on one or more data items, such as adding, multiplying, comparing, or moving.

operator. A symbol that represents the action being performed in a mathematical operation. A person who operates a machine.

operator message. A message from the operating system or a program telling the operator to perform a specific function or informing the operator of a specific condition within the system, such as an error condition.

option. A specification in a statement, a selection from a menu, or a setting of a switch, that may be used to influence the execution of a program. A hardware or software function that may be selected or enabled as part of a configuration process. A piece of hardware (such as a network adapter) that can be installed in a device to modify or enhance device function.

output device. A device in a data processing system by which data can be received from the system. Synonymous with *output unit*.

output unit. Synonym for *output device*.

overflow exception. A condition caused by the result of an arithmetic operation having a magnitude that exceeds the largest possible number. See also *underflow exception*.

P

parameter. A name in a procedure that is used to refer to an argument passed to that procedure. A variable that is given a constant value for a specified application and that may denote the application. An item in a menu or for which the user specifies a value or for which the system provides a value when the menu is interpreted. Data passed between programs or procedures.

parity bit. A binary digit appended to a group of binary digits to make the sum of all the digits (including the appended binary digit) either always odd (odd parity) or always even (even parity).

parity (even). A condition when the sum of all of the digits in an array of binary digits is even.

parity (odd). A condition when the sum of all of the digits in an array of binary digits is odd.

personal computer (PC). A desk-top, free-standing, or portable microcomputer that usually consists of a system unit, a display, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily to give independent computing power to a single user and are inexpensively priced for purchase by individuals or small businesses. Examples include the various models of the IBM Personal Computers, and the IBM Personal System/2 computer.

physical connection. The ability of two connectors to mate and make electrical contact. In a network, devices that are physically connected can communicate only if they share the same protocol. See also *logical connection*.

PLD. Power line disturbance.

PLU. Price Look Up.

plug. A connector for attaching wires from a device to a cable, such as a store loop. A plug is inserted into a receptacle or plug. To insert a connector into a receptacle or socket.

pointer. An identifier that indicates the location of an item of data in memory. A data element that indicates the location of another data element. A physical or symbolic identifier of a unique target.

point-of-sale terminal. A unit that provides point-of-sale transaction, data collection, credit authorization, price look-up, and other inquiry and data entry functions.

polling. Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. In data communication, the process of inviting data stations to transmit, one at a time. The polling process usually involves the sequential interrogation of several data stations.

polling characters (address). A set of characters specific to a terminal and the polling operation; response to these characters indicates to the computer whether the terminal has a message to enter.

port. An access point for data entry or exit. A connector on a device to which cables for other devices such as display stations and printers are attached. Synonymous with *socket*.

post. To affix to a usual place. To provide items such as return code at the end of a command or function. To define an appendage routine. To note the occurrence of an event.

POST. Power-On Self Test.

power line disturbance (PLD). Interruption or reduction of electrical power.

Power-On Self Test (POST). A series of diagnostic tests that are run automatically each time the computer's power is switched on.

problem determination. The process of determining the source of a problem as being a program component, a machine failure, a change in the environment, a common-carrier link, a user-supplied device, or a user error.

procedure. A set of related control statements that cause one or more programs to be performed. In a programming language, a block, with or without formal parameters, whose execution is invoked by means of a procedure call. A set of instructions that gives a service representative a step-by-step procedure for tracing a symptom to the cause of failure.

process. An instance of an executing application and the resources it is using.

processor. In a computer, a functional unit that interprets and executes instructions.

prompt. A character or word displayed by the operating system to indicate that it is ready to accept input.

Q

queue. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message routing system.

R

RAM. Random access memory.

random access. An access mode in which specific logical records are obtained from or placed into a mass storage file in a nonsequential manner.

random access memory (RAM). A computer's or adapter's volatile storage area into which data may be entered and retrieved in a nonsequential manner.

read. To acquire or to interpret data from a storage device, from a data medium, or from another source.

read-only memory (ROM). A computer's or adapter's storage area whose contents cannot be modified by the user except under special circumstances.

receive. To obtain and store information transmitted from a device.

record. A collection of related items of data, treated as a unit; for example, in stock control, each invoice could constitute one record. A complete set of such records form a file.

register. A storage area in a computer's memory where specific data is stored. Registers are used in the actual manipulation of data values during the execution of a program. A storage device having a specified storage capacity such as bit, byte, or computer word, and usually intended for a special purpose. In the IBM Store System, a term that refers to the point-of-sale terminal.

remove. To take an attaching device off a network. To stop an adapter from participating in data passing on a network.

resource. An element that affects the way devices behave.

resource set. The set of resources associated with a device.

response. The information the network control program sends to the access method, usually in answer to a request received from the access method. (Some responses, however, result from conditions occurring

within the network control program, such as accumulation of error statistics.)

retry. In data communication, sending the current block of data a prescribed number of times or until it is entered correctly and accepted.

return code. A value (usually hexadecimal) provided by an adapter or a program to indicate the result of an action, command, or operation. A code used to influence the execution of succeeding instructions. A value established by the programmer to be used to influence subsequent program action. This value can be printed as output or loaded in a register.

ROM. Read-only memory.

routine. Part of a program, or a sequence of instructions called by a program, that may have some general or frequent use.

S

satellite. A computer that is under the control of another computer and performs subsidiary operations. An offline auxiliary computer.

SBCS. Single-byte character set.

scan. To pass an item over or through the scanner so that the encoded information is read. See also *wanding*.

scan codes. When a key is pressed on any keyboard, the keyboard device driver receives a code that is called a *make scan code*. Each key has a different code. When a key is released on some keyboards, the keyboard device driver receives a code that is called a *break scan code*. These codes are translated into ASCII character codes using the code page that the application is using.

scanner. A device that examines the bar code on merchandise tickets, credit cards, and employee badges and generates analog or digital signals corresponding to the bar code.

scroll. To move all or part of the display image vertically or horizontally to display data that cannot be observed within a single display image. See also *page (2)*.

segment. See *cable segment, LAN segment, ring segment*.

sequential file. A disk file in which records are read from or placed into the file according to the order they are processed.

session. A connection between two application programs that allows them to communicate. In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested. The data transport connection resulting from a call or link between two devices. The period of time during which a user of a node can communicate with an interactive system, usually the elapsed time between log on and log off. In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

signal. A time-dependent value attached to a physical phenomenon for conveying data. A variation of a physical quantity, used to convey data.

sign-on. A procedure to be followed at a terminal or workstation to establish a link to a computer. To begin a session at a workstation.

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set.*

SLRS. SUSE LINUX Retail Solution.

socket. Synonym for *port (2)*.

state transition. The act of moving from one conversation state to another.

station. A point-of-sale terminal that consists of a processing unit, a keyboard, and a display. It can also have input/output devices, such as a printer, a magnetic stripe reader or cash drawers. A communication device attached to a network. The term used most often in LANs is an *attaching device* or *workstation*. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line. See also *attaching device, workstation*.

subdirectory. Any level of file directory lower than the root directory within a hierarchical file system.

subroutine. Section of code that performs a specific task and is logically separate from the rest of the program.

subsystem. A secondary or subordinate system, or programming support, usually capable of operating independently of or asynchronously with a controlling system.

summary journal. A record of the terminal operational activity that is printed at the terminal.

switch. On an adapter, a mechanism used to select a value for, enable, or disable a configurable option or feature.

system. In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. See also *data processing system* and *operating system*.

system configuration. A process that specifies the devices and programs that form a particular data processing system.

system unit. A part of a computer that contains the processing unit, and may contain devices such as disk and diskette drives. In an IBM Personal Computer, the unit that contains the processor circuitry, read-only memory (ROM), random access memory (RAM), and the I/O channel. It may have one or more disk or diskette drives. In an IBM 4683/4684 terminal, the part of the terminal that contains the processing unit, ROM, RAM, disk and diskette drives, and the I/O channel.

T

terminal. In data communication, a device, usually equipped with a keyboard and a display, capable of sending and receiving information over a communication channel.

thread. A unit of execution within a process. It uses the resources of the process.

throughput. A measure of the amount of work performed by a computer system over a given period of time, for example, number of jobs per day. A measure of the amount of information transmitted over a network in a given period of time. For example, a network's data transfer rate is usually measured in bits per second.

till. A tray in the cash drawer of the point-of-sale terminal, used to keep the different denominations of bills and coins separated and easily accessible.

touch-down. When contact is made with a touch-sensitive surface.

trace. A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. A record of the frames and bytes transmitted on a network.

transaction. The process of recording item sales, processing refunds, recording coupons, handling voids, verifying checks before tendering, and arriving at the amount to be paid by or to a customer. The receiving of payment for merchandise or service is also included in a transaction. In an SNA network, an exchange between two programs that usually involves a specific set of initial input data that causes the execution of a specific task or job. Examples of transactions include the entry of a customer's deposit that results in the updating of the customer's balance, and the transfer of a message to one or more destination points.

transition. See *state transition*.

transmission. The sending of data from one place for reception elsewhere.

transmit. To send information from one place for reception elsewhere.

typematic. A keyboard button that will continue to enter characters or repeat its function as long as the button is held down.

U

underflow exception. A condition caused by the result of an arithmetic operation having a magnitude less than the smallest possible nonzero number. See also *overflow exception*.

unlock. To enable a device, such as a scanner or MSR, so that it can read data. See also *lock*.

universal product code (UPC). An encoded number that can be assigned to and printed on or attached to an article of merchandise for scanning.

Universal Serial Bus (USB). A serial interface standard for telephony and multimedia connections to personal computers.

Universal Serial Bus (USB), powered. A powered-USB connector provides additional power from the host system. A powered-USB receptacle consists of two connectors stacked vertically inside the common housing. The upper connector contains four contacts that are used for powering the attached device.

UDC. User defined character.

UPC. Universal product code.

user. Category of identification defined for file access protection. A person using a program or system.

user defined character (UDC). User defined character.

user interface. Hardware, software, or both that allows a user to interact with and perform operations on a system, program, or device.

V

variable. A named entity that is used to refer to data and to which values can be assigned. Its attributes remain constant, but it can refer to different values at different times. In computer programming, a character or group of characters that refers to a value and, in the execution of a computer program, corresponds to an address. A quantity that can assume any of a given set of values.

version. A separate IBM-licensed program, based on an existing IBM-licensed program, that usually has significant new code or new function.

W

wanding. Passing the tip of the wand reader over information encoded on a merchandise ticket, credit card, or employee badge.

workstation. An I/O device that allows either transmission of data or the reception of data (or both) from a host system, as needed to perform a job: for example, a display station or printer. A configuration of I/O equipment at which an operator works. A terminal or microcomputer, usually one connected to a mainframe or network, at which a user can perform tasks.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Readers' Comments — We'd Like to Hear from You

Point of Sale Subsystem
UnifiedPOS User's Guide, Keyboards, and Code Pages

Publication No. GC30-4078-12

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
GC30-4078-12

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Retail Store Solutions Information Development
Dept. ZBDA
P.O. Box 12195
Research Triangle Park, NC 27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Updated: March 27, 2009
Printed in USA

GC30-4078-12