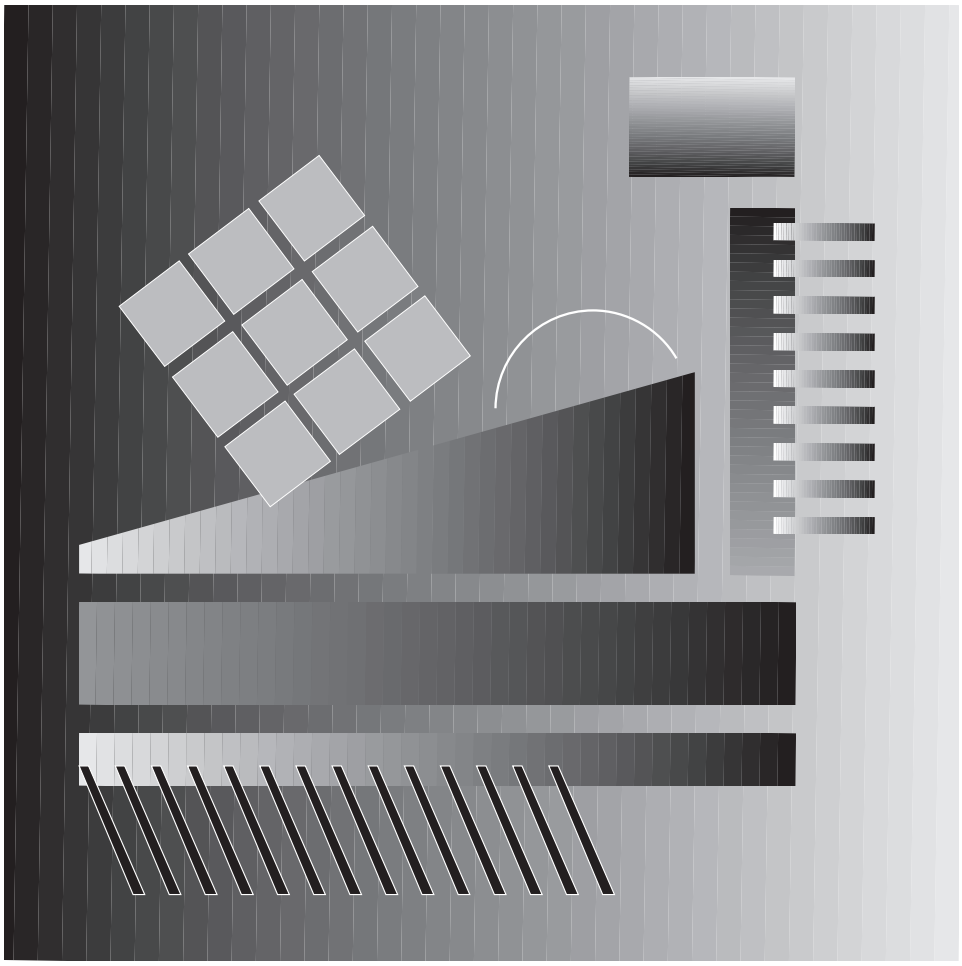
Point of Sale Subsystem

UnifiedPOS User's Guide, Keyboards, and Code Pages



last update: August 29, 2006

GC30-4078-07

Point of Sale Subsystem

UnifiedPOS User's Guide, Keyboards, and Code Pages

last update: August 29, 2006

GC30-4078-07

Notice

Before using this information and the product it supports, be sure to read the general information under Appendix C, “Notices,” on page 363.

Eighth Edition, September, 2006

This edition applies Version 1.9.2 of the UnifiedPOS subsystem and to all subsequent releases and modifications until otherwise indicated in new editions.

Current versions of Retail Store Solutions documentation are available on the IBM Retail Stores Solutions Web site at: 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 provides reference information for programming devices for use with IBM® UnifiedPOS implementations. It also includes information about IBM keyboards and codepages.

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 149 provides the UnifiedPOS properties and methods, and information on resolving problems.
- Part 3, “Keyboards and code pages,” on page 237 provides information about the available keyboards and character sets.

Related publications

The following is a list of related publications. For information about ordering these publications, contact your IBM authorized dealer or marketing representative.

Between major revisions of this manual we might make minor technical updates. The latest softcopy version of this guide is available on the IBM Retail Store Solutions web site:

- Go to www.ibm.com/solutions/retail/store/support.
- Click **Publications**.

UPOS publications

Unified Point of Service (UPOS) Retail Peripheral Architecture:
www.nrf-arts.org/UnifiedPOS/default.htm

Store Systems publications – hardware

Scanners

IBM 1520 Hand-Held Scanner User’s Guide, GA27-3685
IBM 4687 Point of Sale Scanner Model 1: Physical Planning, Installation, and Operation Guide, SA27-3855
IBM 4687 Point of Sale Scanner Model 1: Maintenance Manual, SY27-0317
IBM 4687 Point of Sale Scanner Model 2: Physical Planning Guide, SA27-3882
IBM 4687 Point of Sale Scanner Model 2: Operator’s Guide, SA27-3884
IBM 4687 Point of Sale Scanner Model 2: Maintenance Manual, SY27-0324
IBM 4696 Point of Sale Scanner: Maintenance Manual, SY27-0333
IBM 4696 Point of Sale Scanner: Physical Planning, Installation, and Operation Guide, GA27-3965
IBM 4697 Point of Sale Scanner Model 001: Maintenance Manual, SY27-0338
IBM 4697 Point of Sale Scanner Model 001: Physical Planning, Installation, and Operation Guide, GA27-3990

IBM 4698 Point of Sale Scanner Scale Model 001 & 002: Physical Planning, Installation, and Operation Guide, GA27-4055
IBM 4698 Point of Sale Scanner Scale Model 001 & 002: Maintenance Manual, SY27-0344

Cabling

A Building Planning Guide for Communication Wiring, G320-8059
IBM Cabling System Planning and Installation Guide, GA27-3361
IBM Cabling System Catalog , G570-2040
Using the IBM Cabling System with Communication Products, GA27-3620

IBM 4610 SureMark Point of Sale printer

IBM 4610 SureMark™ Point of Sale printer: User's Guide, GA27-4151

IBM 4694 Point of Sale Terminals

IBM Store Systems: Technical Reference, SY27-0336
IBM 4694/4695 Point of Sale Terminal: Hardware Service Manual, SY27-0337
IBM Store Systems: Hardware Service Manual for Point of Sale Terminal Input/Output Devices, SY27-0339
IBM 4694 Point of Sale Terminal: User's Guide, SA27-4005
IBM 4694 Point of Sale Terminal: Hardware Service Manual, SY27-0364
IBM Store Systems: Installation and Operation Guide for Point of Sale Input/Output Devices, GA27-4028
IBM Store Systems: Point of Sale Terminals – Supplement for Installation, Operation, and Service, GA27-4035

IBM SurePOS 500/600 Series

IBM SurePOS™ 500/600 Planning, Installation, and Operation Guide, GA27-4254
IBM SurePOS 500/600 System Reference, SA27-4255
IBM SurePOS 500/600 Hardware Service, GY27-0396

IBM SurePOS 300

IBM SurePOS 300 Installation and Service Guide, GA27-4309
IBM SurePOS 300 Technical Reference
IBM Kiosk Model 120 System Reference, SA27-4289

SurePOS 700 Series

IBM SurePOS 720, 740, and 780 Planning, Installation and Operation Guide, GA27-4328
IBM SurePOS 720, 740, and 780 Hardware Service Guide, SA27-4329
IBM SurePOS 700 Series: Installation and Operation Guide, GA27-4293
IBM SurePOS 700 Series: Hardware Service Manual, GY27-0363
IBM SurePOS 700 Series: System Reference, SA27-4224
IBM SurePOS 700 Series: Options and I/O Devices Service Guide SY27-0392

IBM 4820 SurePoint Solution

IBM 4820 SurePoint™ Solution: Installation and Service Guide, GY27-4231
IBM 4820 SurePoint Solution: System Reference, SA27-4249

IBM 7497 Point of Sale Attachment Adapter

Point of Sale Terminal Attachment Kit: Physical Planning, Installation, and Service Manual, GA27-4034

IBM Point of Sale Subsystem

Point of Sale Subsystem Programming Reference and User's Guide, SC30–3560

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: www.ibm.com/solutions/retail/store

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

Tell us what you think

Your feedback is important in helping us provide accurate and high-quality information. Please take a few moments to tell us what you think about this document.

You can use either of these ways to provide feedback:

- Visit our Web site at http://www.ibm.com/solutions/retail/store. Select **Support**, then **Publications**. Select the **publication comments** link within the introductory text. Fill in the requested information and your comments. Be sure to include the name and form number of the document in the Publication ID field.
- Print and fill out the form at the end of this document. Return the form to us by mail or by giving it to an IBM representative.

If applicable, include a reference to the specific location of the text (for example, the page or table number) on which you are commenting.

Between major revisions of this document, we might make minor technical updates. The latest version of this document is available on the Retail Store Solutions Web site at www.ibm.com/solutions/retail/store/support/publications.

Summary of changes

Eighth edition (GC30-4078-07, September, 2006)

This edition includes UPOS 1.9.2.

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

Seventh edition (GC30-4078-06, April, 2006)

This edition includes UPOS 1.9.1.

Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341 has been added.

Sixth edition (GC30-4078-05, November, 2005)

This edition includes UPOS 1.9.

In this edition Line Scanner and Omni Scanner are added.

Properties and methods tables are updated.

Fifth edition (GC30-4078-04, July 2005)

This edition includes new information about UPOS 1.7.5.

Web-only update (January 2005)

This update contains new or additional information related to the Anyplace Kiosk (4836/4838).

Fourth edition (GC30-4078-03, December 2004)

This edition contains new or additional information about OLE for Retail Point of Sale (OPOS).

Third edition (GC30-4078-02, September 2004)

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

Second edition (GC30-4078-01, May 2004)

This edition includes information about 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 1.9.2 includes driver packages for both JavaPOS and 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. The IBM OPOS driver set includes a technology demonstration called the *OPOS Gateway*. The 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, thus 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 1.9.2 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)	
4694 POS Terminal	2x5, 2x6, 2x7, 3x7
	OPOS only: 0x1, 0x4, Sx1, Sx4, 1x4, 2x4
SurePOS 300 Series	
SurePOS 500/600 Series	
SurePOS 700 Series	
Kiosk	
SureOne®	

RS-485 and PS/2 attached Point of Sale devices

The following RS-485 and PS/2® attached Point of Sale devices are supported:

Cash drawers:

- No till (feature code 3360)
- Adjustable till (feature code 1092)
- Fixed till (feature code 3879)

- Cash drawer, removable till (feature code 3361)
- Flip-top cash drawer (feature code 3362)
- Cash drawer I (P/N 6238669)
- Cash drawer IV (P/N 09F3519)
- Cash drawer V (Feature code 3370)
- Compact Cash drawer with vertical till (feature code 3368)
- Compact cash drawer with horizontal till (feature code 3378)

Displays:

- Shopper Display (Feature code 3339)
- Operator Display (Feature code 3340)
- 40-Character Alphanumeric Display (Feature code 3343) (OPOS only)
- 50-Key Modifiable Layout Keyboard/Operator Display (Feature code 6300)
- Character/Graphics Display (Feature code 3400)
  - Japan:
    - Tall (Feature code 3402)
    - Short (Feature code 3403)
  - Korea:
    - Tall (Feature code 3405)
    - Short (Feature code 3406)
- PLU Keyboard/Display - III
  - Japan (Feature code 3230)
  - Korea (Feature code 3232)
- 40-Character Vacuum Fluorescent Display II (Feature code 3501)
- 40-Character Vacuum Fluorescent Display II - Japan (Feature code 3506)
- Two-sided Vacuum Fluorescent Display II (Feature code 3502)
- Two-sided Vacuum Fluorescent Display II - Japan (Feature code 3507)
- 40-Character Liquid Crystal Display (Feature code 3503)
- Retail Point of Sale Keyboard with Card Reader and Display (Feature code 6300)
- 2x20 Character Vacuum Fluorescent Display Customer Display (Feature code 2826)

Keyboards:

- 50-Key Modifiable Keyboard (Feature code 3320)
- 50-Key Modifiable Layout Keyboard/Operator Display (Feature code 6300)
- Alphanumeric Point of Sale Keyboard (NANPOS) (Feature code 3324)
- Compact Alphanumeric Point of Sale Keyboard (CANPOS)
- Retail Alphanumeric Point of Sale Keyboard
  - Brazil/Portuguese (Feature code 3200)
  - Danish (Feature code 3211)
  - Canada/French (Feature code 3201)
  - French (Feature code 3203)
  - German (Feature code 3204)
  - Italian (Feature code 3205)
  - Norwegian (Feature code 3212)
  - Spanish (Feature code 3206)

  - Swedish/Finnish (Feature code 3213)
  - US English (Feature code 3324)
  - UK English (Feature code 3202)
- Retail Alphanumeric Point of Sale Keyboard with Card Reader
  - Brazil/Portuguese (Feature code 3200)
  - Danish (Feature code 3211)
  - Canada/French (Feature code 3201)
  - French (Feature code 3203)
  - German (Feature code 3204)
  - Italian (Feature code 3205)
  - Norwegian (Feature code 3212)
  - Spanish (Feature code 3206)
  - Swedish/Finnish (Feature code 3213)
  - US English (Feature code 3324)
  - UK English (Feature code 3202)
- Retail Point of Sale Keyboard (Feature code 3315)
- Retail Point of Sale Keyboard with Card Reader (Feature code 3320)
- Modifiable Layout Keyboard with Card Reader (Feature code 3323)
- 4820 SurePoint Solution Keypad
- PC Point of Sale Keyboard
  - Japan (Feature code 3207)
  - Korea (Feature code 3208)
- Point of Sale Keyboard V
  - Japan (Feature code 3220)
  - Korea (Feature code 3221)
- Point of Sale Keyboard VI - Korea (Feature code 3209)
- PLU Keyboard Display III
  - Japan (Feature code 3230)
  - Korea (Feature code 3232)
  - Attached to 4674 with expansion box
- 4674 Point of Sale Keyboard (built-in)
- Retail Point of Sale Keyboard with Card Reader and Display (Feature code 6300)
- 4685 Point of Sale Keyboard Model K01 (4685-K01) (OPOS only)
- 4685 Point of Sale Keyboard Model K02 (4685-K02)
- 4685 Point of Sale Keyboard Model KC1 (4685-KC1) (OPOS only)
- 4685 Point of Sale Keyboard Model K02 Ultra 7 with MSR/E
- 4685 Point of Sale Keyboard Model K02 Ultra 7

Magnetic stripe readers:

- One-Track Magnetic Stripe Reader (ISO Track 2) (Feature code 4010)
- Dual-Track Magnetic Stripe Reader (ISO Tracks 1 and 2) (Feature code 4192)
- Dual-Track Magnetic Stripe Reader (ISO Tracks 2 and 3) (Feature code 4193)
- Low Profile Dual-Track Magnetic Stripe Reader (ISO Tracks 1 and 2) (Feature code 6310)
- Low Profile Dual-Track Magnetic Stripe Reader (ISO Tracks 2 and 3) (Feature code 6320)

- Retail Alphanumeric Point of Sale Keyboard with Card Reader
  - Brazil/Portuguese (Feature code 3200)
  - Danish (Feature code 3211)
  - Canada/French (Feature code 3201)
  - UK English (Feature code 3202)
  - French (Feature code 3203)
  - German (Feature code 3204)
  - Italian (Feature code 3205)
  - Norwegian (Feature code 3212)
  - Spanish (Feature code 3206)
  - Swedish/Finnish (Feature code 3213)
  - US English (Feature code 3324)
- Retail Alphanumeric Point of Sale Keyboard
  - Brazil/Portuguese (Feature code 3200)
  - Danish (Feature code 3211)
  - Canada/French (Feature code 3201)
  - UK English (Feature code 3202)
  - French (Feature code 3203)
  - German (Feature code 3204)
  - Italian (Feature code 3205)
  - Norwegian (Feature code 3212)
  - Spanish (Feature code 3206)
  - Swedish/Finnish (Feature code 3213)
  - US English (Feature code 3324)
- Retail Point of Sale Keyboard with Card Reader (Feature code 3320)
- Modifiable Layout Keyboard with Card Reader (Feature code 3323)
- PC Point of Sale Keyboard
  - Japan (Feature code 3207)
  - Korea (Feature code 3208)
- Point of Sale Keyboard V
  - Japan (Feature code 3220)
  - Korea (Feature code 3221)
- PLU Keyboard Display III
  - Japan (Feature code 3220)
  - Korea (Feature code 3221)
- Point of Sale Keyboard VI - Korea (Feature code 3209)
- Retail Point of Sale Keyboard with Card Reader and Display (Feature code 6300)
- Compact Alphanumeric Point of Sale (CANPOS) Keyboard
- Three-track Magnetic Stripe Reader (Feature code 2905)
- Two-sided Magnetic Stripe Reader (Feature code 2906)
- SurePoint Magnetic Stripe Reader (Feature code 3951)
- SurePoint JUCC Magnetic Stripe Reader (Feature code 3953)
- 4685 Point of Sale Keyboard Model K01 (4685-K01) (OPOS only)
- 4685 Point of Sale Keyboard Model K02 Ultra 7
- 4685 Point of Sale Keyboard Model K02 Ultra 7 with MSR/E

- 4685 Point of Sale Keyboard Model KC1 (OPOS only)
- 4693 Point of Sale Terminal Model 202 (4693-202)
- 4693 Point of Sale Terminal Model 212 (4693-212)
- 4693 Point of Sale Terminal Model 2S2 (4693-2S2)

Nonvolatile random access memory:

- 4674 Point of Sale Terminal (Japan only) Models: 001, 011, 010, and DS1 (OPOS only)
- 4674 Point of Sale Terminal (Japan only) Model 121
- 4694 POS Terminal Models 0x1, 0x4, Sx1, Sx4, 1x4, 2x4 (OPOS only)
- 4694 POS Terminal Models 2x5, 2x6, 2x7, 3x7
- SurePOS 700
- SureOne (except A04/A05) (OPOS only)

Printers:

- IBM Model 3 Printer (Feature code 4700) (OPOS only)
- IBM Model 3Fx/3Bx Printer (JavaPOS only)
- IBM Model 4 Printer (Feature code 4800) (OPOS only)
- IBM Model 4A Printer (Feature code 4805) (OPOS only)
- IBM Model 4R Printer (Feature code 4801) (OPOS only)
- 4610 SureMark Point of Sale Printers Models: TI1, TI2, TI3, TI4, TI5, TF6, TF7, TM6, TM7, TI8/TG8, TI9/TG9, TG3, TG4, Gx3, Gx5
- 4610 SureMark Point of Sale Printers Models: Kx3, Kx4, Kx5 (JavaPOS Only)
- 4689 Point of Sale Printer Models:
  - 001 - Japan (Feature code 4802) (OPOS only)
  - 002 - Korea (Feature code 4803) (OPOS only)
  - 301 (4689-301)
  - 3G1 (Japan only) (OPOS only)
  - 3M1 (Japan only) (OPOS only)
  - TD5 (Japan only)

Scales:

- 4687 Point of Sale Scanner Model 002
- 4696 Point of Sale Scanner Scale Model 001
- 4698 Point of Sale Scanner Model 002, 101, 102

Scanners:

- IBM Hand-Held Bar Code Reader Models:
  - IBM Hand-Held Bar Code Reader Model 1 (Feature Code 4500)
  - IBM Hand-Held Bar Code Reader Model 2 (Feature Code 4501)
- IBM 1520 Hand-Held Scanner Model A02 (1520-A02)
- 4685 Point of Sale Scanner Model LOG
- 4685 Hand-Held Bar Code Reader Models:
  - 001 (Feature code 4502)
  - L01 (Handy Scanner III)
  - L0A (Japan only)
- 4685 Point of Sale Scanner Model L0F (Japan only)

- 4685 SurePOS Scanner (Asia Pacific country) Models: S01, L0C, L0H, 101
- 4687 Point of Sale Scanner Models: 001, 002
- 4696 Point of Sale Scanner Scale Model 001
- 4697 Point of Sale Scanner Model 001
- 4698 Point-of-Sale Scanner Models: 001, 002, 201 (Japan only)

Check scanners:

- 4610 TI8 and TI9

SurePOS 300 devices

The following SurePOS 300 devices are supported:

Table 2. SurePOS 300 devices supported by UnifiedPOS

Cash Drawers	Full-size cash drawer (fixed till)
	Full-size cash drawer (adjustable till)
	Compact cash drawer (horizontal till)
	Compact cash drawer (vertical till)

SurePOS 500/600 devices

The following SurePOS 500/600 devices are supported:

Table 3. SurePOS 500/600 devices supported by UnifiedPOS

Displays	Integrated Display
Cash Drawers	Full-size cash drawer (fixed till)
	Full-size cash drawer (adjustable till)
	Compact cash drawer (horizontal till)
	Compact cash drawer (vertical till)
Magnetic stripe readers	ISO Reader
	JUCC Reader
Motion sensor	Presence Sensor

SureOne devices

The following SureOne devices are supported:

Table 4. SureOne devices supported by UnifiedPOS

Displays	Remote Display
Cash Drawers	Full-size cash drawer (fixed till)
	Full-size cash drawer (adjustable till)
	Compact cash drawer (horizontal till)
	Compact cash drawer (vertical till)
Magnetic stripe readers	1-2 Track Reader
	2-3 Track Reader
Printers	Impact Printer
	Thermal Printer
Keyboards	POS System Keyboard
Keylock	3 Position keylock

4674 Devices

The following 4674 devices are supported:

Table 5. 4674 devices supported

Displays	Integrated
Cash Drawers	4674 Cash Drawer
Keyboards	4674 Keyboard Keylock and Tone Indicator JUCC MSR
Printer	Integrated 4689 TD5 Printer

Kiosk devices

The following Kiosk devices are supported:

Table 6. Kiosk devices supported by UnifiedPOS

Displays	Integrated Display
	Remote Display
	Remote APA Display
Magnetic stripe readers	ISO Reader
	JUCC Reader
Motion sensor	Presence Sensor
Scanner	IBM Anyplace Line Scanner
	IBM Anyplace Omni Scanner

USB Point-of-Sale devices

The following USB Point-of-Sale devices are supported:

Table 7. USB Point-of-Sale devices supported by UnifiedPOS

Keyboards	50-key keyboard
	50-key keyboard with magnetic stripe reader (MSR)
	50-key keyboard with magnetic stripe reader (MSR) and Liquid Crystal Display (LCD)
	Alphanumeric point-of-sale (NANPOS) keyboard
	133-key keyboard with MSR
	4685 Point of Sale Keyboard Model K02 (4685-K02)
Displays	40-character vacuum fluorescent display (VFD), one-sided
	40-character VFD, two-sided
	40-character LCD
Cash Drawers	Full-size cash drawer (fixed till)
	Full-size cash drawer (adjustable till)
	Compact cash drawer (horizontal till)
	Compact cash drawer (vertical till)

Table 7. USB Point-of-Sale devices supported by UnifiedPOS (continued)

Printers	4610 SureMark Point of Sale Printers Models: TI3, TI4, TI5, TF6, TF7, TM6, TM7, TI8, TI9, Gx3, Gx5, Kx
	Model 4 Printer with Protocol Converter
	4689 Point of Sale Printer
Miscellaneous	OEM scale and scanners
	Standard USB devices (such as a mouse)

Note:

IBM does not support hot-plugging of powered USB (12v/24v) devices on 72x/74x/78x SurePOS systems.

EIA-232 attached devices

The following EIA-232 attached devices are supported:

Table 8. EIA-232 attached devices supported by UnifiedPOS

Displays	Remote Display
	Remote APA Display
	IBM 2x20 distributed character display
Cash Drawers	Full-size cash drawer (fixed till)
	Full-size cash drawer (adjustable till)
	Compact cash drawer (horizontal till)
	Compact cash drawer (vertical till)
Printers	4610 SureMark Point of Sale Printers Models: TI3, TI4, TI5, TF6, TF7, TM6, TM7, TI8, TI9, Kx, Gx

SurePoint (4820) display

The SurePoint Display supports the following connection types: PS/2, EIA-232 (MSR only), RS-485, and USB.

Table 9. SurePoint devices

Keyboards	32-key keypad
Magnetic Stripe Readers	ISO Reader
	JUCC Reader

Software environment

The IBM UnifiedPOS system requires the following software environment:

Table 10. Software environment for IBM UPOS

Operating system	One of the following: • Microsoft® Windows 2000 Service Pack 4 • Microsoft Windows XP Service Pack 2 • Microsoft Windows Embedded Point of Service (WEPOS) • SuSE Linux® Retail Solution 8 (SLRS) (JavaPOS only) • Novell Linux Point of Service (NLPOS)

Table 10. Software environment for IBM UPOS (continued)

JVM	IBM SDK or JRE 1.4.2 build IBM SDK or JRE142SR2 or later. (Determine the build level of SDK or JRE by entering `java -fullversion`.)

The OPOS Common Control objects are required to run the IBM OLE for Retail Subsystem. These objects can be obtained from http://www.monroecs.com/oposccos.htm. IBM does not provide control objects with its installation.

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 Setup.exe 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 or both of the following features:
   - OPOS
   - JavaPOS

   Be sure to inspect the sub-features for correct system unit support.

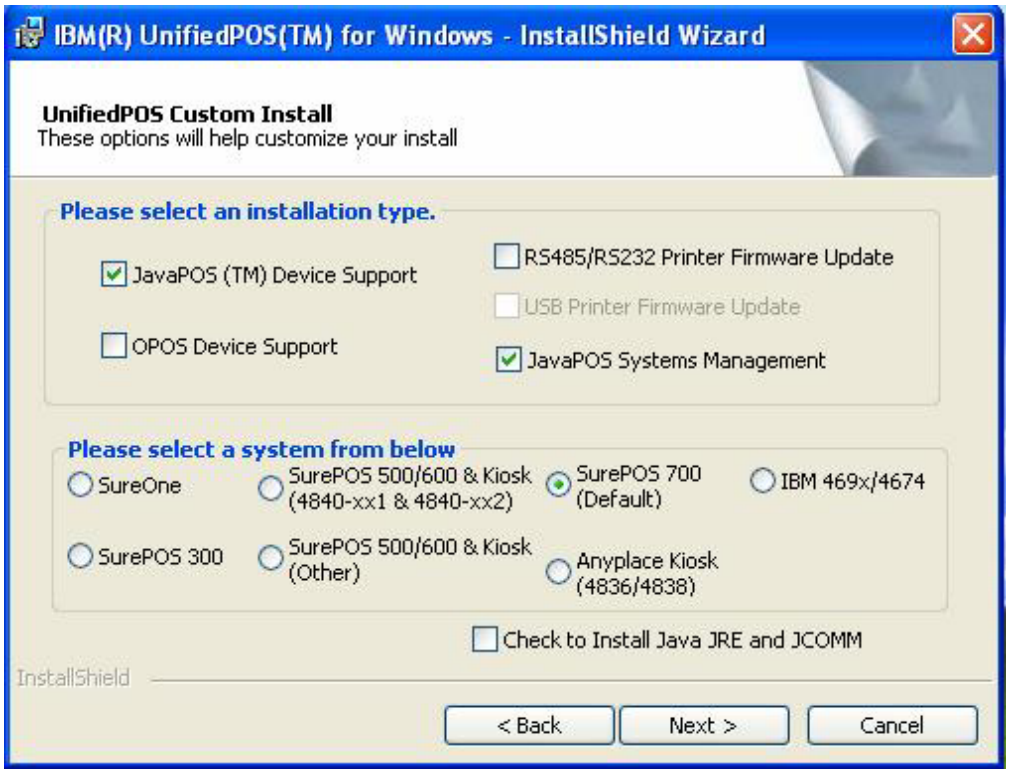


Figure 1. Features dialog

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

Silent installation

IBM UnifiedPOS can be installed and updated silently (unattended) using a response file. The response file, <system root drive>\UposJreSetup.iss for the JRE

installer and <system root drive>\UposJdkSetup.iss for the SDK installer, 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.

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

Unattended installation with reboot

To perform an unattended installation or update, use the following command:

```
setup /s /v/qn
```

To enable logging for the installation, use the following command:

```
setup /s /v"/qn /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 /s /v"/qn REBOOT=A"
setup /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 double quotes are required as shown.
3. The file, C:\msilog.txt contains the install log.

Uninstallation

Uninstallation can be performed interactively using **Add/Remove Programs** in the Windows Control Panel.

To perform an unattended uninstall, use the following command:

```
setup /s /x /v/qn
```

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

For 4840 systems:

Prior to 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 thus the IBM 4840 System Port), when you reboot, the hardware manager will see a new device which is the IBM 4840-xx3 PCI Multi-I/O Controller. The device manager will prompt you to reload the drivers for this device.

Installation for IRES

JavaPOS drivers are included with IRES and have been tested with SUSE Linux Retail Solution 8 (SLRS 8) and Novell Linux Point of Service 9 (NLPOS 9). To understand how the JavaPOS drivers are integrated into IRES, refer to the IRES Developer's Guide at `http://www2.clearlake.ibm.com/store/support/html/pubs.html`.

For more information about SLRS 8 and the NLPOS 9, visit `http://www.suse.com` and `http://www.novell.com/documentation/nlpos9`.

Chapter 3. Configuring devices

This chapter covers configuration for JavaPOS devices, OPOS devices, and USB access.

Configuring devices for JavaPOS

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

This section describes how to create a jpos.xml file.

Locating sample jpos.xml files

During the installation, the following sample XML files are installed on your system:

Linux

`/opt/ibm/javapos/docs/jpos.xml.LinuxSample`

Windows

`/pos/ibmjpos/docs/jpos.xml.WinSample`

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

In your new jpos.xml file, keep only those `JposEntries` that correspond to devices on the target system.

Creating 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 jpos.xml to pos\ibmjpos\config directory in Windows, or to /opt/ibm/jpos/etc in Linux.
6. Update CLASSPATH if jpos.xml is saved in another directory.

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 directory to the location of the jpos.xml file.
3. Open the JavaPOS Configuration Editor by typing the following:

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

Note: The JavaPOS Configuration Editor can also be accessed from Programs->IBM JavaPOS->JavaPOS Configuration Editor. For more information go to: `http://sourceforge.net/projects/jposloader/`

4. Change the Logical Name in each JposEntry to match the Logical Name for that device in your JavaPOS application.

Changing the logical name of JposEntry: Follow these steps to change the logical name of JposEntry:

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

Note: To conserve resources, jpos.xml must contain JposEntries only for those devices attached to the system. Delete all other JposEntries.

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.

Configuring EIA-232 Entries: The entries for the EIA-232 devices are in the `jpos.xml` file. Some contain specific properties for configuring communication with the device. The specific values accepted for each of them are documented in Table 11.

Table 11. EIA-232 device communication properties

JCL property name	Type	Possible values (typical values)	Default Value
portName	String	Valid port name such as: COM1	"" - empty
baudRate	String	"75", "110 ", "134", "150", "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"

Note: 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″.

Values for the device communication properties for specific EIA-232 devices are shown in Table 12. The *dataBits*, *parity* and *stopBits* properties are omitted from the table and from jpos.xml entries. The default values shown in Table 11 on page 22 are used for those properties.

Notes:

1. Values different than those described here will cause the device to fail.
2. In the following table, ″Is omitted″ means that the user does not have to type these values into the xml.

Table 12. Specific Values for EIA-232 devices in JavaPOS.

Device	portName	baudRate	flowControl
CashDrawer 4610-A CashDrawer 4610-B	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200" (depends on configuration of the printer)	Only "Xon/Xoff"
CashDrawer 4840-A CashDrawer 4840-B	Default: "COM4"	Default: "9600" Is omitted	"None" Is omitted
CheckScanner 4610-TI8/TI9	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200" (depends on configuration of the printer)	Only "Xon/Xoff"
Fiscal Printer 4610-Kx3 Fiscal Printer 4610-Kx4 Fiscal Printer 4610-Kx5	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200"	Only "None"
Line Display 4840 APA	Default: "COM4"	"9600" Is omitted	"Hardware"
Line Display 4810/4840/SureOne VFD	Default: "COM4"	"9600" Is omitted	"None" Is omitted
MICR 4610 TI2/4/8/9	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200" (depends on configuration of the printer)	Only "Xon/Xoff"
MSR 4820/4836/4838/4840 ISO MSR 4830/4840 JUCC	Default: "COM3"	"19200"	"None" Is omitted
POSPrinter 4610 TI1/2/3/4/8/9 TM6/7, TF6/7	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200" (depends on configuration of the printer)	Only "Xon/Xoff"
POSPrinter SureOne	Only "COM3"	"9600"	Only "Xon/Xoff"
IBM Anyplace Kiosk Line Scanner	Default: "COM4"	"19200"	"None" Is omitted

Table 12. Specific Values for EIA-232 devices in JavaPOS. (continued)

Device	portName	baudRate	flowControl
IBM Anyplace Kiosk Omni Scanner	Default: "COM4"	Default: "9600"	"None" Is omitted
Tone Indicator 4610 TM/TF 6/7	Port where the printer is attached. Default: ″COM1″	Default: "9600" "19200" (depends on configuration of the printer)	Only "Xon/Xoff"

Note: For the devices embedded in the 4610 printer, the entry of the device and the printer must be configured in order for the device to work.

POS Control Center utility

To access the IBM POS Control Center:

- In Windows, select **Programs** > **IBM JavaPOS** > **POS Control Center**
- In Linux, enter `POSControlCenter`

The POS Control Center displays devices found on the system and device entries from the jpos.xml file, along with device information. The device information is helpful if discrepancies occur between properties specified in JposEntry and actual properties of the attached device. 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.

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

The 4810-x3x POS hardware has an embedded setting which shows that there is a second cashdrawer, as well as 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 should not be configured in the jpos.xml file as valid devices.

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

JavaPOS Auto-Configuration

Before describing the design and operation of the Auto-Configuration code, a review of the JavaPOS API and implementation is required. The JavaPOS stack is as shown in the following graphic.

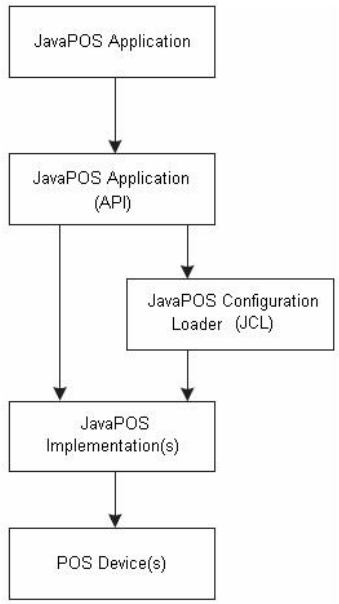


Figure 2. JavaPOS stack

When the JavaPOS application calls open() on a JavaPOS control (such as POSPrinter.open), the JavaPOS API accesses the JavaPOS Configuration Loader (JCL) which has a table of all configured devices. Entries in this table are accessed by logicalName. The table is initialized on the first access, during JCL initialization. The JCL is designed for completely static operation: it loads a predefined static list of entries from a file called “jpos.xml”. This user-created file defines exactly what devices were connected to the system when it was created. If the JavaPOS control finds the entry with the logicalName specified by the user, it connects the control to the implementation.

The Auto-Configuration code changes the operation of the JCL by redefining the classes that are used to initialize the JCL table.

The new JCL implementation instead of reading from the jpos.xml file, initializes the IBM JavaPOS driver stack, which is used to detect all physically connected IBM POS devices. At the end, the JCL table is composed of those devices that are physically connected to the system.

Limitations

The Auto-Configuration does as much automatic configuration as possible, however some configuration must be done manually. For example, almost all EIA-232 devices require you to provide additional configuration, as EIA-232 devices. Also the 4610 cash drawers(all buses) are not auto-detectable.

Also, some devices have user-configurable parameters that are set to default values but can be changed by editing the specific device properties file.

Each of the device properties files has a list of all the configurable properties along with the allowed values that can be used.

Auto-config properties files

1. posj.autoconfig.<device>. properties files

   The configuration of the IBM JavaPOS drivers is done using properties files that can be found in Linux /opt/ibm/javapos/etc

   The following list shows the available properties for every device.

Table 13.

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

2. 2. posj.autoconfig.properties

   Controls the implementation class for some of the Auto-Configuration code. The defaults should be acceptable for all users and this file should not be modified. jpos/res/jpos.properties controls the JCL operation. There are two modes of JCL operation:

   - Auto-configuration only mode

     This mode creates a JCL table with entries created only by the Auto-Configuration code. For this mode, the jpos.loader.serviceManagerClass and the jpos.config.regPopulatorClass should be redefined:

     ```
     #jpos.loader.serviceManagerClass=jpos.loader.simple.SimpleServiceManager
     jpos.loader.serviceManagerClass=com.ibm.jpos.tools.autoconfig.FixedManager
     #jpos.config.regPopulatorClass=jpos.config.simple.SimpleRegPopulator
     jpos.config.regPopulatorClass=com.ibm.jpos.tools.autoconfig.SDIPopulator
     ```

   - jpos.xml and Auto-Configuration

     This mode creates a JCL table with all entries from a jpos.xml file and all entries created by the Auto-Configuration code. Note that if the Auto-Configuration code tries to create an entry with a logical name already used by a jpos.xml entry, the jpos.xml entry takes precedence and the Auto-Configuration entry is discarded. To enable this mode the jpos.loader.serviceManagerClass and the jpos.config.regPopulatorClass should be redefined:

```
#jpos.loader.serviceManagerClass=jpos.loader.simple.SimpleServiceManager
jpos.loader.serviceManagerClass=com.ibm.posj.autoconfig.FixedManager
#jpos.config.regPopulatorClass=jpos.config.simple.SimpleRegPopulator
jpos.config.populator.class.0=jpos.config.simple.xml.SimpleXmlRegPopulator
jpos.config.populator.class.1=com.ibm.jpos.tools.autoconfig.SDIPopulator
```

Enabling Auto-Config

JavaPOS includes some sample jpos.properties files that will enable a specific mode just by making a copy of the file to the first directory available in your CLASSPATH. The copied file should be located in the following path: jpos\res\jpos.properties.

The sample file is jpos.properties.autocfg and can be found in Linux /opt/ibm/javapos/docs.

Enabling XML and Auto-Config mode

To enable XML and Auto-Config mode make a copy of the sample configuration file to the first directory available in your CLASSPATH. The file should be located in the following path jpos\res\jpos.properties. The sample file is jpos.properties.xmlautocfg and can be found in Linux /opt/ibm/javapos/docs

Disabling Auto-Config

To disable any Auto-Config mode make a copy of the sample configuration file to the first directory available in your CLASSPATH. The file should be located in the following path jpos\res\jpos.properties. The sample file is jpos.properties.xmlcfg and can be found in Linux /opt/ibm/javapos/docs

Device specific notes

4610 Printer Cash Drawers

The printers attached to the 4610 can not be automatically detected even when they are connected to the printer. To enable any of the cash drawers, the file posj.autoconfig.POSPrinter.properties contains 2 properties that enable any of the cash drawers. By default the properties are commented and disabled:

```
#autoconfig.hasCashDrawerA.1=false
#autoconfig.hasCashDrawerB.1=false
```

To enable the properties, they should be commented out and enabled:

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

EIA-232 settings

Because EIA-232-connected devices are not usually auto-detectable, you must specify some or all of the EIA-232 settings. The settings must be specified in number-specific format and set in any of the posj.autoconfig.<device>.properties files. EIA-232 settings can not be specified in default format. (Refer to the section Autoconfiguration properties files format) At a minimum, the EIA-232 portName and deviceBus must be specified. Additionally, the baudRate and flowControl can be set also. deviceBus must be set to RS232.

Note: EIA-232-connected CashDrawer is the only auto-detectable EIA-232 device, so CashDrawers on the SurePOS 500 using IRES2 systems do not need to have their EIA-232 settings specified. However, their settings may be specified if desired.

MICR Exception Handling

The way Auto-Config handling of exception table entries is through a list of entries saved in a file, one entry per line. The following is an example of how this file may look like:

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

Then the file posj.autoconfig.MICR.properties should be edited to specify the name of the file containing MICR exceptionTable entries. The name of the property is:

```
autoconfig.exceptionTableFile=micrentries.txt
```

Autoconfiguration properties files format

This is a general description of the posj.autoconfig.<device type>.properties files. There are to different parts that should be understood: *format* and *logicalName handling*.

Format: All of the properties in the posj.autoconfig.<device>.properties files, can be specified in two formats:

1. Default format

   property=value
   where;
   property = Any of the device properties available for the device.
   value = Any valid value to be used to initialized the property

2. Number-specific format

   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.
   value = Any valid value for the property

   The setting of the property using this format will apply only to the device identified by number.(see “Device enumeration order” on page 29), so the rest of devices of the category will keep its default value unless a property is explicitly set.

Example 1

To enable the property stripTransitDashes for all of the MICR devices auto-detected in the system, the default format should be used:

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

Example 2

If only the first MICR device should enable the flag, then the number-specific format should be used also, indicating that only the first device found will use this value (the rest will use the default value)

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

logicalName property considerations: Because the logicalName is used as the key to add entries into the JCL table, there can only be one entry with any particular logicalName. Therefore, it can only be set using the number-specific 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 first CashDrawer logicalName would be CashDrawer1 and the third LineDisplay would be LineDisplay3. These defaults can be overridden in each posj.autoconfig.<device>.properties file.

EIA-232 section: All of the posj.autoconfig.<device type>.properties that can be configured in EIA-232 mode include a special section related to EIA-232 settings. The EIA-232 device contains at least the following properties, and it can set some other device specific properties.

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

Propertes baudRate and flowControl affects only the POSPrinter device.

Device enumeration order

The JavaPOS drivers enumerate POS devices in a predefined order. This order can not be changed and the actual order of device enumeration must match the device configuration described in this section. The enumeration order is as follows:

- System Keyboards
  - PS/2 keyboards (and their integrated devices)
  - USB “system” mode keyboards (only, not integrated devices)
- USB devices

  Enumeration proceeds from low-numbered USB ports to high-numbered USB ports within a specific USB hub. Across USB hubs (internal or external) or Host Controllers, enumeration order depends entirely on the system ordering of the Host Controllers and the connection order (low port to high port) of the hubs.
- “Embedded” devices

  This includes many (but not all) system CashDrawers, HardTotals, and MotionSensors.
- RS-485 devices
  - Type 5C Touch Screen
  - Type 5D Touch Screen
- • MSRs
  - MSR 3-track
  - MSR 1-track
  - MSR 2-track
- LineDisplays
  - APA Display (A and B)
  - Alphanumeric Display
  - APA Display (C and D)
- Printers
  - Model 3 Printer
  - 4610 Printer
  - 4689 Thermal Printer
  - Model 2 Printer
  - 4689 Printer
  - Fiscal 3F Printer
- Scanners

  - Flat-bed Scanner
  - Hand-Held Scanner
  - Aruba Scanner
- Keyboards
  - Checkout Keyboard
  - Text Keyboard
  - Matrix Keyboard
  - PLU Keyboard
  - ANPOS Keyboard
  - POS Keyboard
- EIA-232 devices

  Enumerated in order defined.

Tools

There is a tool that will show in html format all the devices that were detected by Auto-Config, along with the assigned logicalName and properties. The instruction is:

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

The output will be a list of all the devices, and the result of the operations Open/Claim/Enable. The following image shows the resulting panel:

last update: August 29, 2006

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

Figure 3. Device list

The following image shows the description created for the device MSR, which can be through the logicalName link “AutoConfigMSR1”:

AutoConfigMSR1		
Property Name	**Property Value**	**Property Type**
abstractionClass	com.ibm.jpos.services.IBMMSR	class java.lang.String
com.ibm.jpos.tools.autoconfig.devcat.DevCatAutoConfig.devCatNumber	1	class java.lang.String
com.ibm.posj.bus.deviceNumber	0	class java.lang.String
com.ibm.posj.bus.rs485.sioDeviceNumber	0x48	class java.lang.String
com.ibm.posj.bus.rs485.sioPortNumber	0x11	class java.lang.String
com.ibm.posj.bus.rs485.sioSlotNumber	0x1	class java.lang.String
deviceBus	RS485	class java.lang.String
deviceCategory	MSR	class java.lang.String
impClass	com.ibm.jpos.services.sdi.MSRServiceImp	class java.lang.String
jposVersion	1.9.140	class java.lang.String
logicalName	AutoConfigMSR1	class java.lang.String
productDescription	MSR	class java.lang.String
productName	MSR	class java.lang.String
productURL	http://www.ibm.com/solutions/retail/store/	class java.lang.String
serviceClass	com.ibm.jpos.services.IBMMSR	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 4. Device list

Supported devices

The JavaPOS Auto-Configuration code supports a subset of devices supported by the JavaPOS drivers. The following list presents the supported devices.

- Cash Drawers
  - SurePOS 72x/74x/78x
  - SurePOS 730/750
  - SurePOS 4840/4851-5x1/5x2/5x3
  - SurePOS 3xx
  - 469x (2x5/2x6/2x7/3x7)
  - 4610 Printer-attached
  - 4674
  - SureOne
- LineDisplays
  - 2x20 VFD (USB/EIA-232/RS-485)
  - 2x20 2-sided VFD (USB/RS-485)
  - 2x20 LCD (USB/RS-485)
  - 50-key keyboard LCD (USB/RS-485)

  - APA, DBCS (USB/EIA-232/RS-485)
  - PLU keyboard APA (USB/RS-485)
- HardTotals
  - 4694
  - 4674
  - SurePOS 730/750
  - SurePOS 72x/74x/78x
- POSKeyboards
  - 50-key keyboard (USB/RS-485)
  - NANPOS keyboard (USB/RS-485/PS2)
  - 120-key keyboard (USB/RS-485)
  - Keyboard V (RS-485)
  - PLU keyboard (USB/RS-485)
  - ANKPOS keyboard (USB/RS-485/PS2)
  - Keyboard 4685 with MSRE Ultra VII (RS-485)
  - Keyboard 4685 (RS485, USB)
  - CANPOS keyboard (PS2)
  - SurePoint keypad (USB/EIA-232/RS-485/PS2)
  - SureONE Keyboard
  - 4674 K03 (USB/RS485)
- MotionSensors
  - 4835-x5x
  - 4836
  - 4838
  - 4840-563/5W3/573/15W
- POSPrinters
  - 4610-TI1/TI2 (EIA-232/RS-485)
  - 4610-TI3 (USB/EIA-232/RS-485)
  - 4610-TI4 (USB/EIA-232/RS-485)
  - 4610-TI8 (USB/EIA-232/RS-485)
  - 4610-TM6/TF6/DM6/IF6 (USB/EIA-232/RS-485)
  - 4610-TI5/TG5 (EIA-232/RS-485)
  - 4610-TM7/TF7 (EIA-232/RS-485)
  - 4689 (RS-485/USB)
  - SureOne (EIA-232)
  - 4610-TI9 (USB/EIA-232/RS-485)
- FiscalPrinters
  - 4610-KC4/KC5/KD3/KC5/FV5 (RS-485/EIA-232)
  - 4610-Gx3/Gx4/Gx5 (USB/RS-485)
  - 2A, 3A, 3BM, 3BS,3F2, 3F, 3FA, 3FB (RS-485)
- Scanners
  - 4500/4501 (RS-485)
  - 1520-A02 (RS-485)
  - 4685-001/K01 (RS-485)
  - 4696 Model 1 (RS-485)

  - 4697 Model 1 (RS-485)
  - 4698 Model 1/2 (RS-485)
  - 4683 (EIA-232)
  - OEM Scanners (USB/RS-485)
- Scales
  - 4696 Model 1 (RS-485)
  - 4698 Model 2 (RS-485)
  - OEM USB Scales (USB)
- CheckScanners (integrated into POSPrinter - see POSPrinters)
- MICRs (integrated into POSPrinter - see POSPrinters)
- Keylocks (integrated into POSKeyboard - see POSKeyboards)
- MSRs (integrated into POSKeyboard - see POSKeyboards)
- ToneIndicators (integrated into POSKeyboard - see POSKeyboards)

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

USB device access

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 usbfs filesystem 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 filesystem permissions must be changed to allow non-root access. There are several ways to do this. All ways involve adding mount parameters to the usbfs filesystem. To manually add parameters, you can edit the /etc/fstab file and add the parameters to the fourth field, or you can manually remount the filesystem and provide the parameters to the mount program.

The most secure way is to change the ownership of the USB device nodes to the JavaPOS application's user. This allows only the JavaPOS application's user to access the USB devices on the system. To do this, mount the usbfs filesystem with 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, use `devuid=1000`. When the filesystem is mounted, all USB device nodes will be owned by `javaposapp`, instead of `root`, so `javaposapp` will be able to access all USB POS devices through the JavaPOS drivers. 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 do this is to create a special group for the users who run the JavaPOS application. In this case, two parameters must be used in mounting usbfs: `devgid=n`, where *n* is the numeric id of the group to which the users who will run the JavaPOS application belong, and `devmode=0664`, which changes the permissions of the USB device nodes to `read/write by owner and group`. This allows any user in the specified group to access all USB devices on the system. 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 the JavaPOS application 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.

Windows 2000

Currently, Windows 2000 has a conflict with USB 2.0 and IBM Retail devices. To circumvent this problem, it is suggested that USB 2.0 be disabled in the System BIOS settings.

Configuring devices for OPOS

The IBM OPOS 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 and use a value different from the assigned default, or can even allow you to specify some resource values.

Configuring your applications

The OPOS configuration window has a separate tab for each of the four POS Terminal types supported by the IBM OPOS package (see Figure 5).

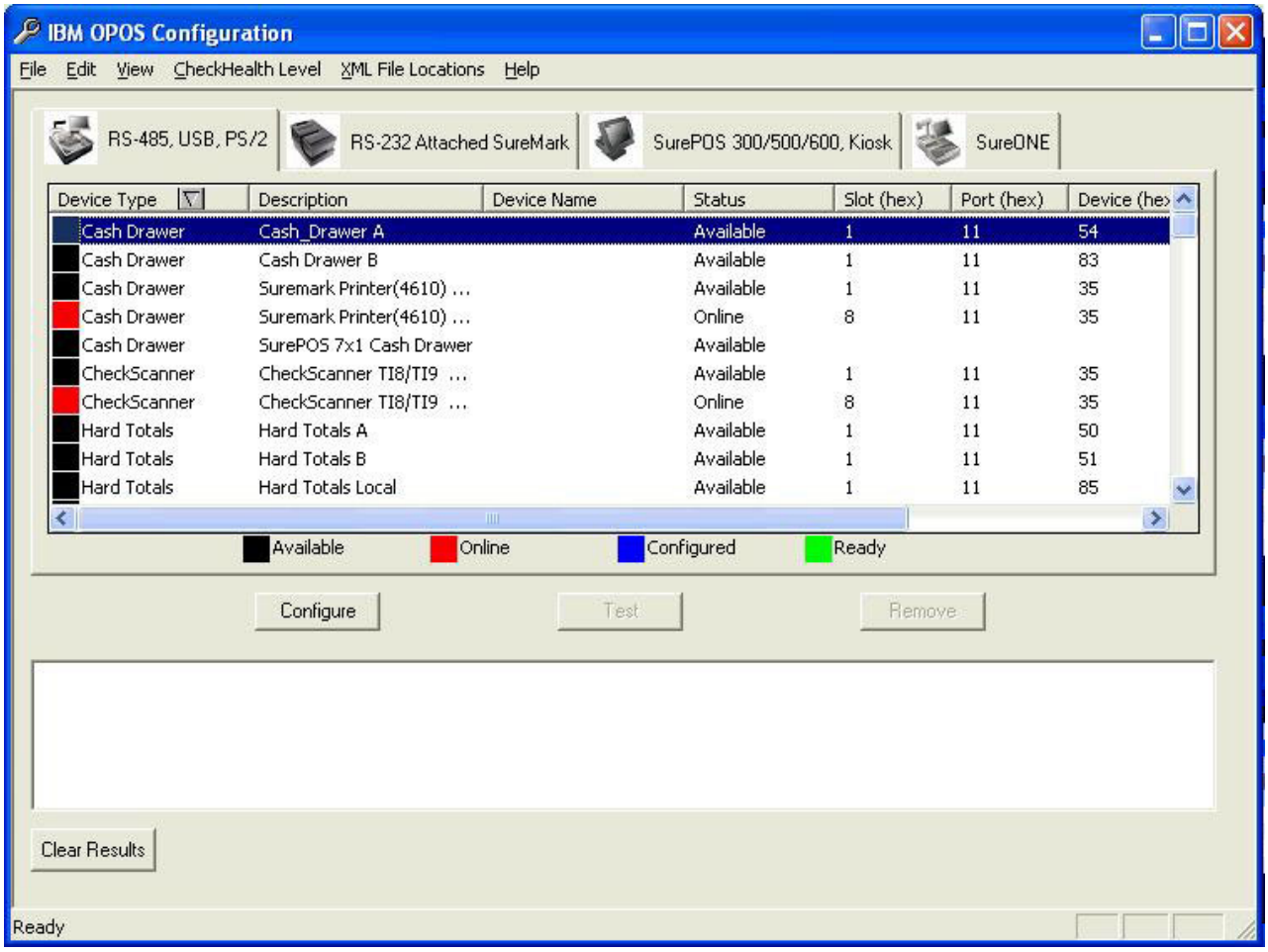


Figure 5. OPOS configuration window

Each tab contains a list of devices that are supported on the POS Terminal. Each device contains a list of settings as well as a status for each of the entries. The status shows what level of configuration is completed using a description, and a color code (see Figure 6 on page 37). Table 14 on page 37 explains the meaning of each status type.

Figure 6. Device status color codes

Table 14. 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 because it uses the system speaker. Other devices can be listed as Online after testing.

Configuration

To configure a device, perform the following steps:

1. Select an entry in the device list. The **Configure** button is enabled.
2. Clicking the button displays a configuration dialog (see Figure 7 on page 38). The dialog differs depending on the POS Terminal type and the device type. Figure 7 on page 38 shows the configuration dialog for a RS-485 device. The slot, port, and device number are displayed. The device number is disabled because that number is specific to the type of device, and changing it could change the device type. The configuration utility only lists some of the more common slot and port settings, but these can be modified for configuration on other POS terminals or feature cards. The supported options are listed in the combo boxes for the fields.

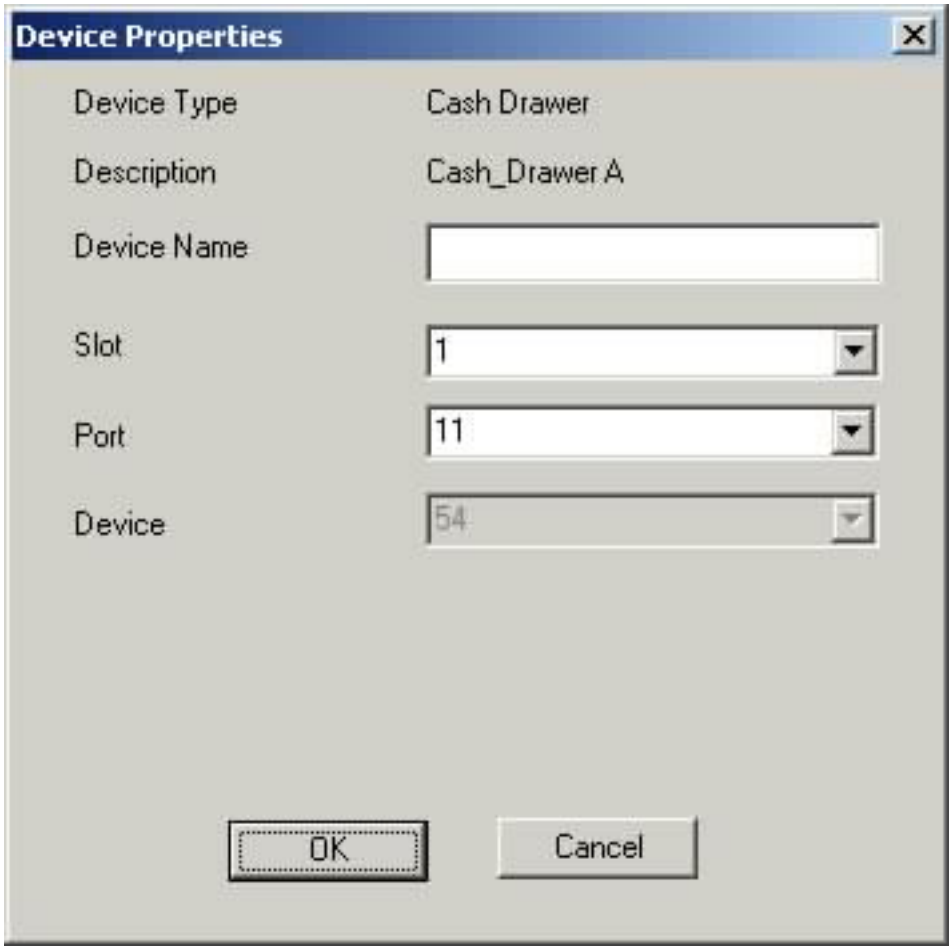


Figure 7. Device properties dialog

3. Enter a name for the device and click **OK**. The entry is added to the device configuration list.

Note: Devices that are not attached can be configured.

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 8 on page 39).

Figure 8. Results of connectivity test

Entry removal

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

Navigation

Navigating the different POS terminal types is as simple as clicking on the tabs at the top. 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.

Deployment

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

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.

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 9 on page 40.)

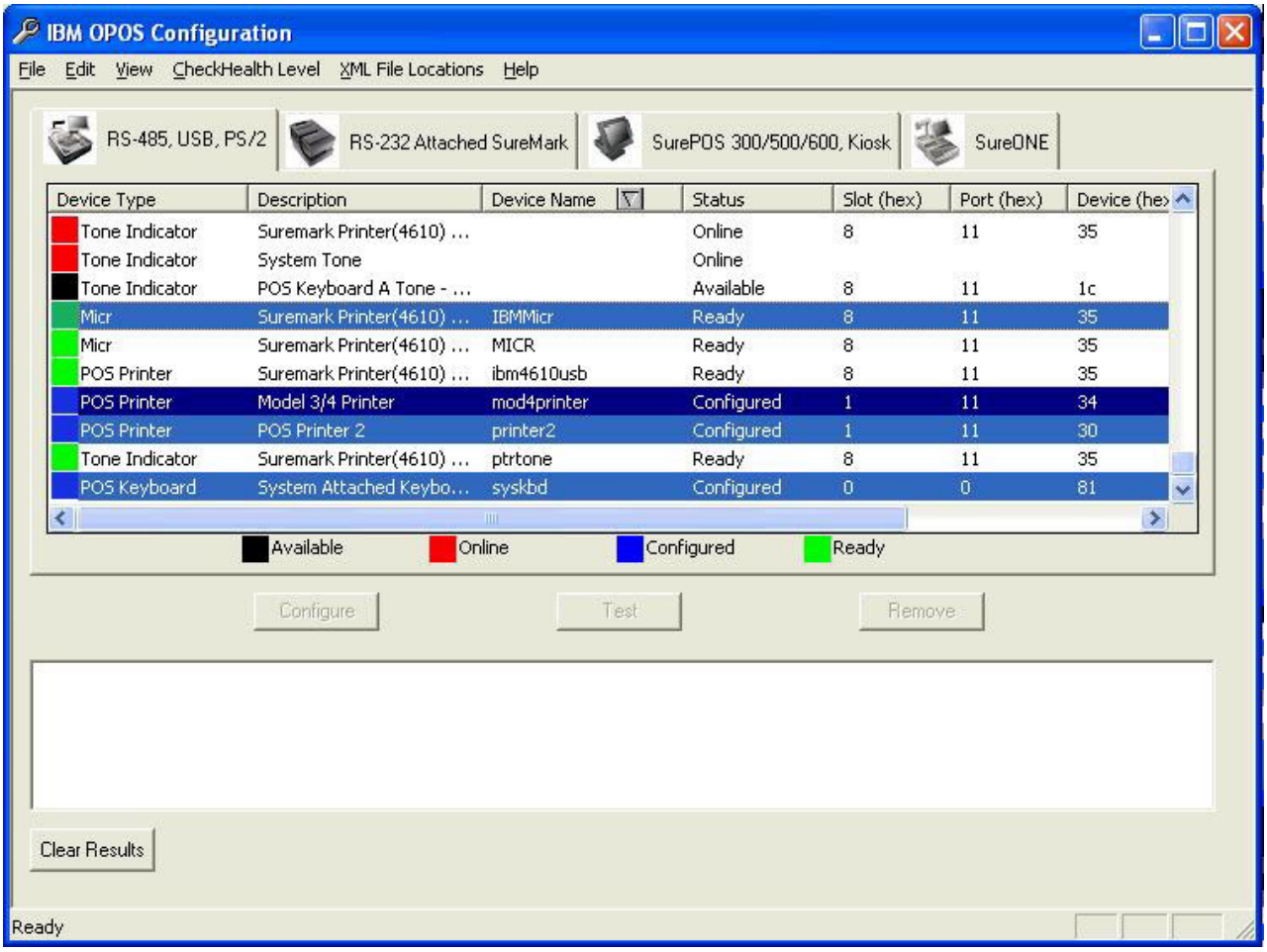


Figure 9. Selecting devices to export

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.

Keyboard configuration

Keyboard and keyboard mapping are fully integrated. 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 10 on page 41).

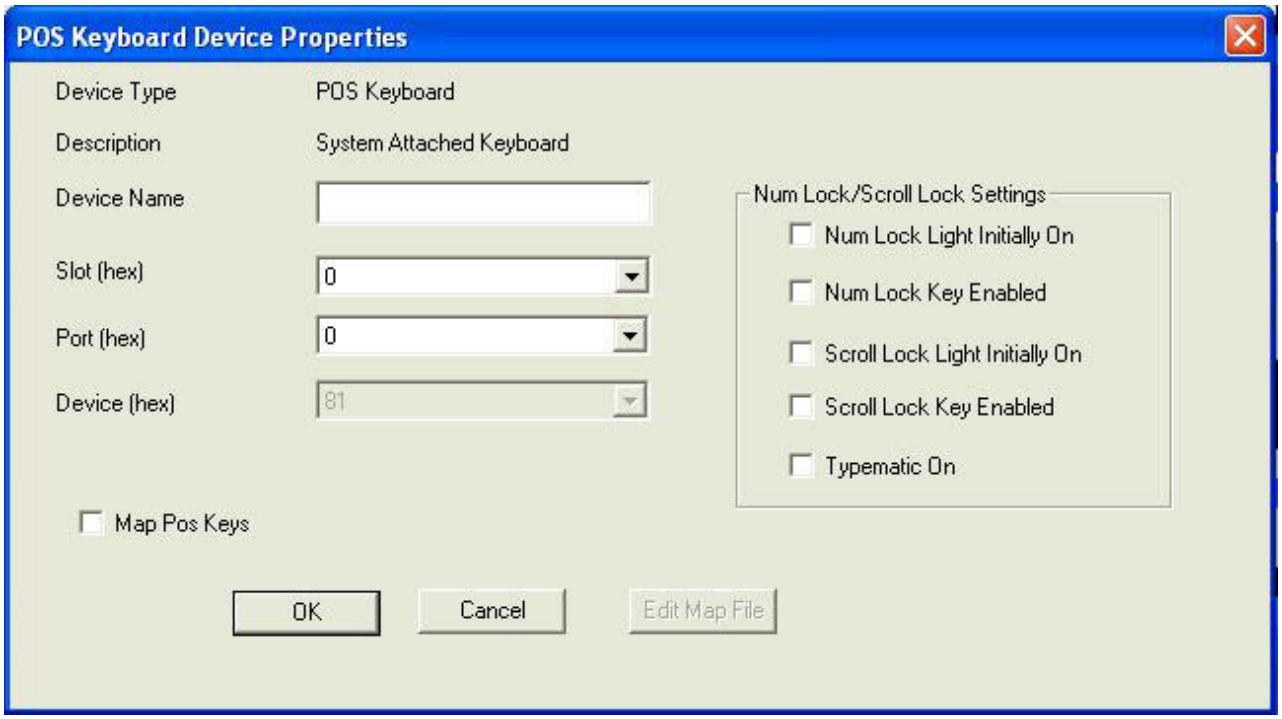


Figure 10. Keyboard properties dialog

2. Select or clear the **Num Lock** and **Scroll Lock** to specify the initial settings you want for those keys.
3. To configure POS keyboard mapping, select **Map Pos Keys** and click the **Edit Map File** button. A filename field is displayed.
4. 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 11 on page 42).

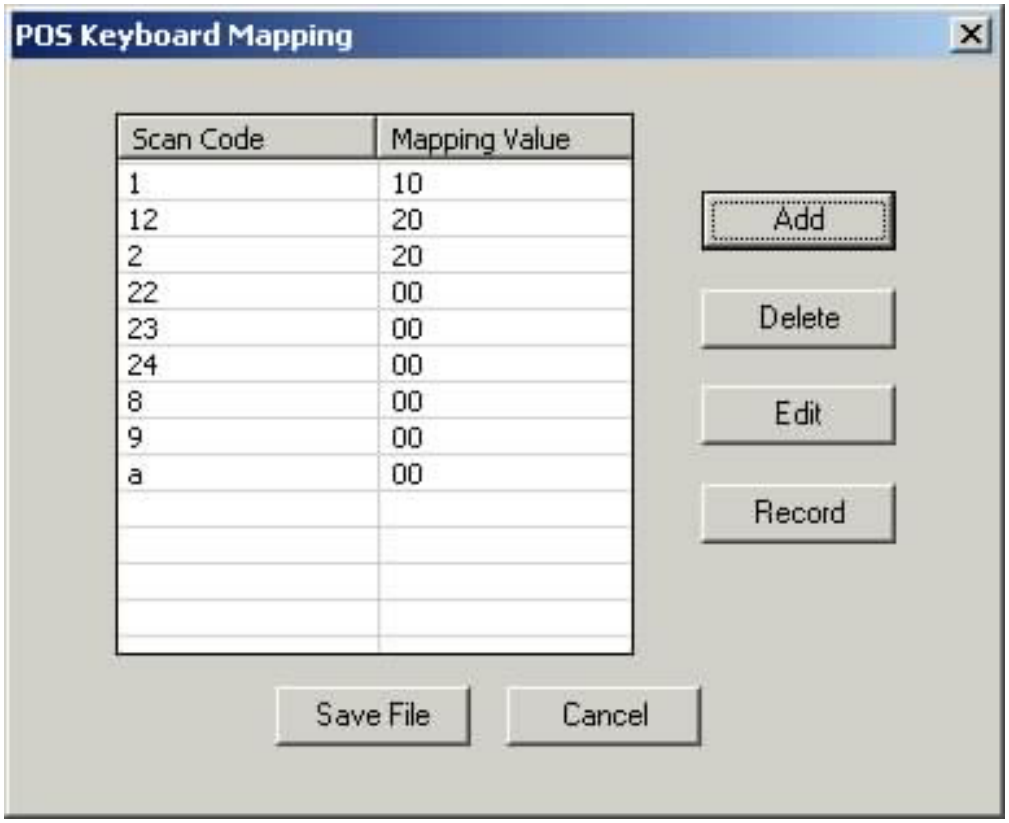


Figure 11. Sample POS keyboard mapping dialog

5. 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**.

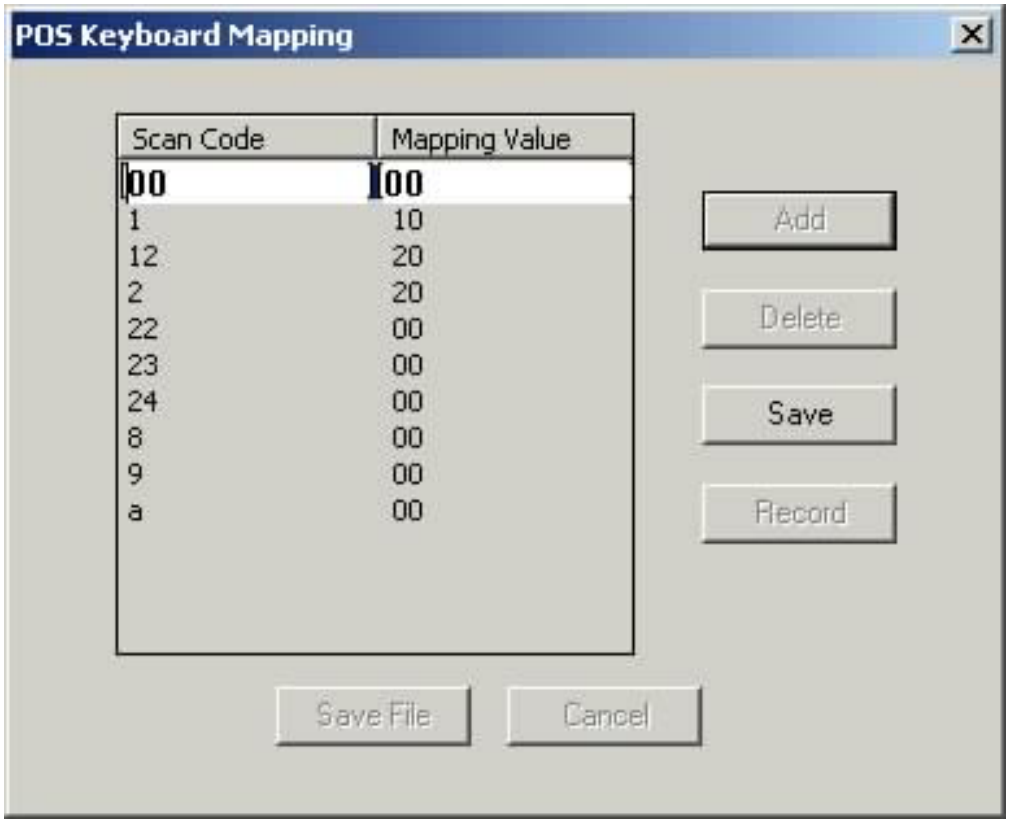


Figure 12. Example of adding a keyboard mapping entry

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**.

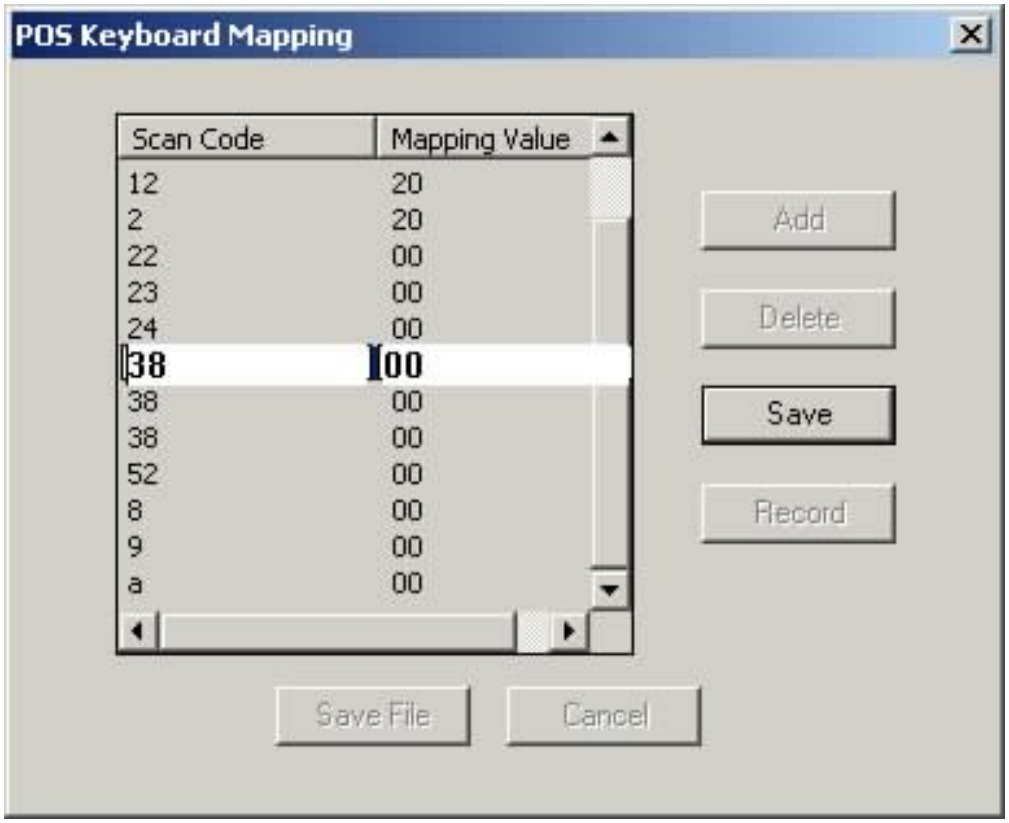


Figure 13. Editing a keyboard mapping entry

- 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 **Edit** to assign mapping values to the new entries.

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

Configuring the Alphanumeric Point of Sale keyboard

If your system has an alphanumeric point-of-sale keyboard attached as the Point of Sale keyboard, you should use the ANPOS utility program to:

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

The ANPOS utility runs automatically at boot time. It uses the resource file, aipsys.res, which is included with the IBM Point of Sale Subsystem.

- On OS/2 and Windows, aipsys.res is located in the IBM Point of Sale root directory (default C:\POS).
- On Linux systems, aipsys.res is located in the /etc directory.

On OS/2, the ANPOS utility program can be run from the STARTUP.CMD file or from an OS/2 command prompt. On Windows 3.1, the ANPOS utility program can be run from the WIN.INI file or from the Run prompt under Program Manager. The ANPOS utility program requires as a parameter the fully-qualified path and file name of a resource file containing the resources that are to be overridden. For example, to use the resource file C:\POS\AIPSYS.RES as a parameter to the ANPOS utility program, the following command would be used:

```
C:\POS\BIN\AIPANPOS C:\POS\AIPSYS.RES
```

A sample resource file named AIPSYS.RES is included in the default C:\POS directory. If you installed the IBM Point of Sale Subsystem in a different location, substitute the correct directory name for C:\POS.

Notes:

1. Use the ANPOS utility program to set the double keys and the numeric keypad zero on an alphanumeric point-of-sale keyboard that is attached as the Point of Sale system keyboard. Requests from an application using the IBM Point of Sale Subsystem subroutines are ignored.
2. The definitions of the keyboard characteristics must use the application name aipanpos and the device name system. This allows them to be specified in the resource file used by your application. For example:

   ```
   aipanpos.system.keyboardClick:     SOFT
   aipanpos.system.doubleKey01        77,82
   aipanpos.system.doubleKey02        90,95
   ```
3. The ANPOS utility records any errors in the file, aipanpos.log. On OS/2 and Windows systems, aipanpos.log is created in the LOG directory of the IBM Point of Sale Subsystem root directory (default C:\POS\LOG). On Linux systems, aipanpos.log will be created in the /var/log directory.
4. The AIPANPOS utility has been modified to support the USB system keyboard (there is no SIO system keyboard). The main functions of this utility are:
   - Double key table download
   - LED initial setting
   - Key click setting

   The POS LEDs and the key click setting are not supported for the USB System Keyboard. There is no change in the function of the Point of Sale keyboard that is attached to the system keyboard port.
5. This information does not apply to the CANPOS keyboard.

CheckScanner configuration

Notes:

1. A POS machine can not have both RS-485 and USB checkscanners operating at the same time.
2. For a CheckScanner, the printer must be Model TI8 or TI9.

To configure an RS-485 CheckScanner, select the entry as shown in Figure 14, and click **Configure**. In the Device Properties popup, add a device name.

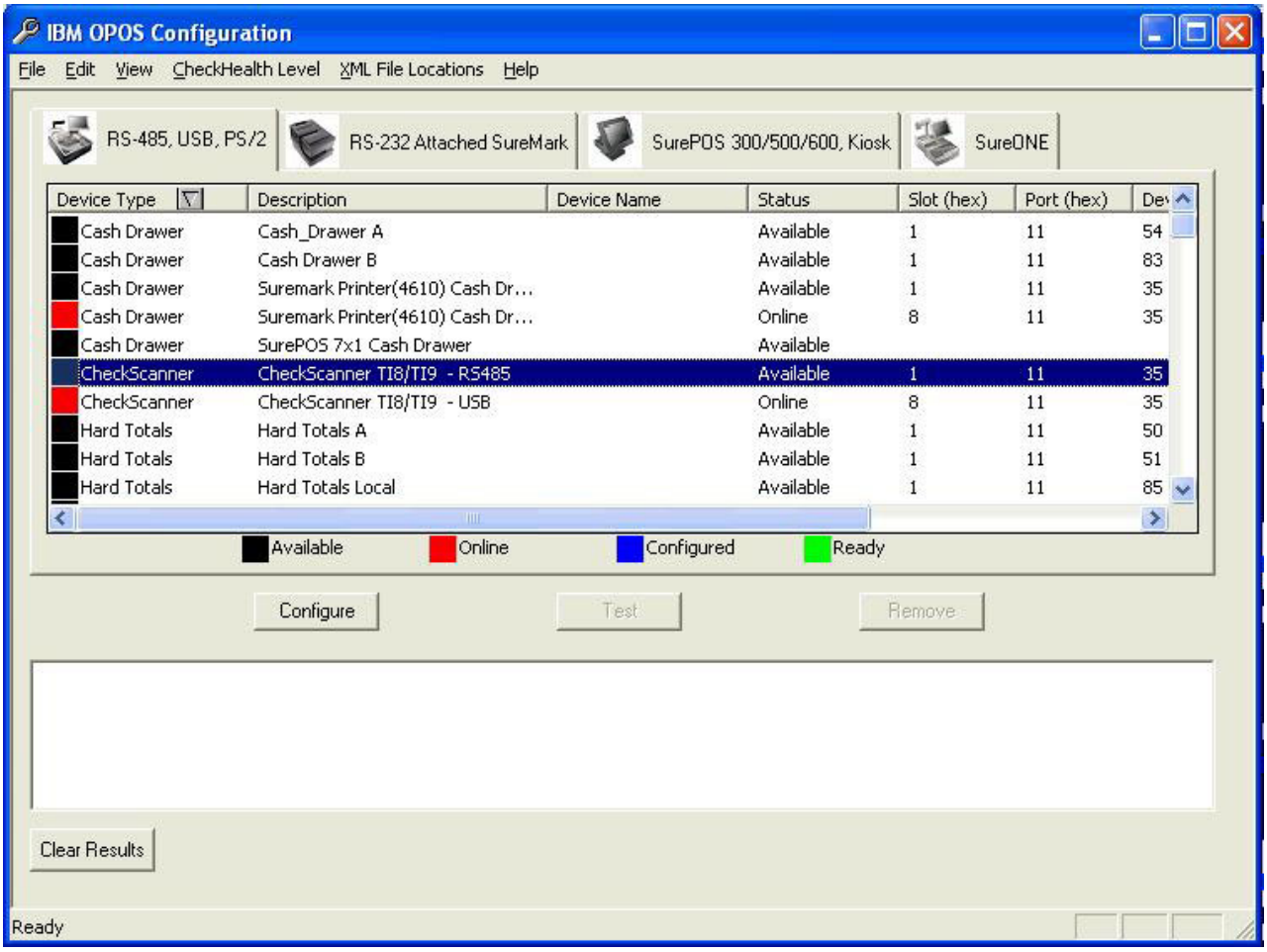


Figure 14. CheckScanner configuration panel, RS-485 device

Figure 17 on page 49 shows CheckScanner1 in the Device Name field.

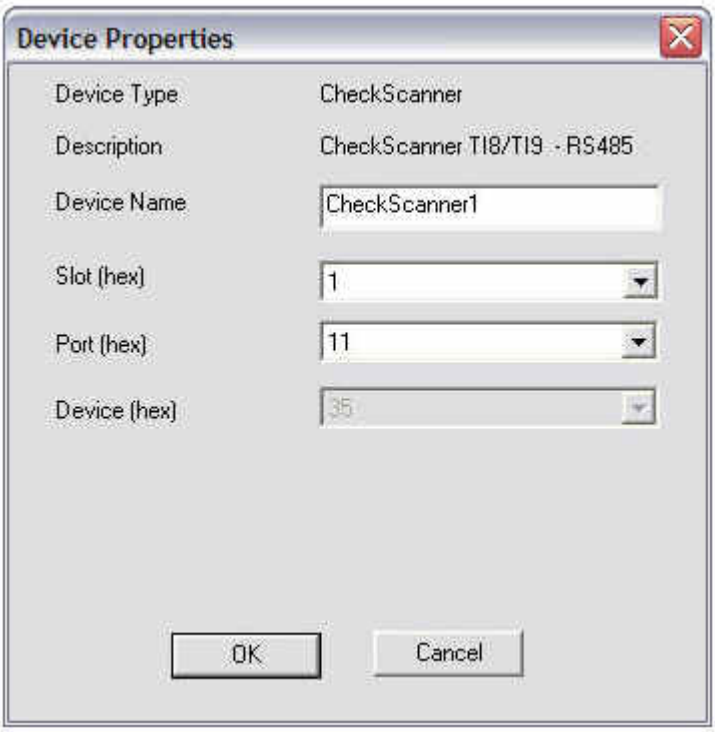


Figure 15. CheckScanner device properties, RS-485 device

Figure 16 shows a USB CheckScanner selected. Note that the Slot number is 8, the Port number is 11 and the Device is 35.

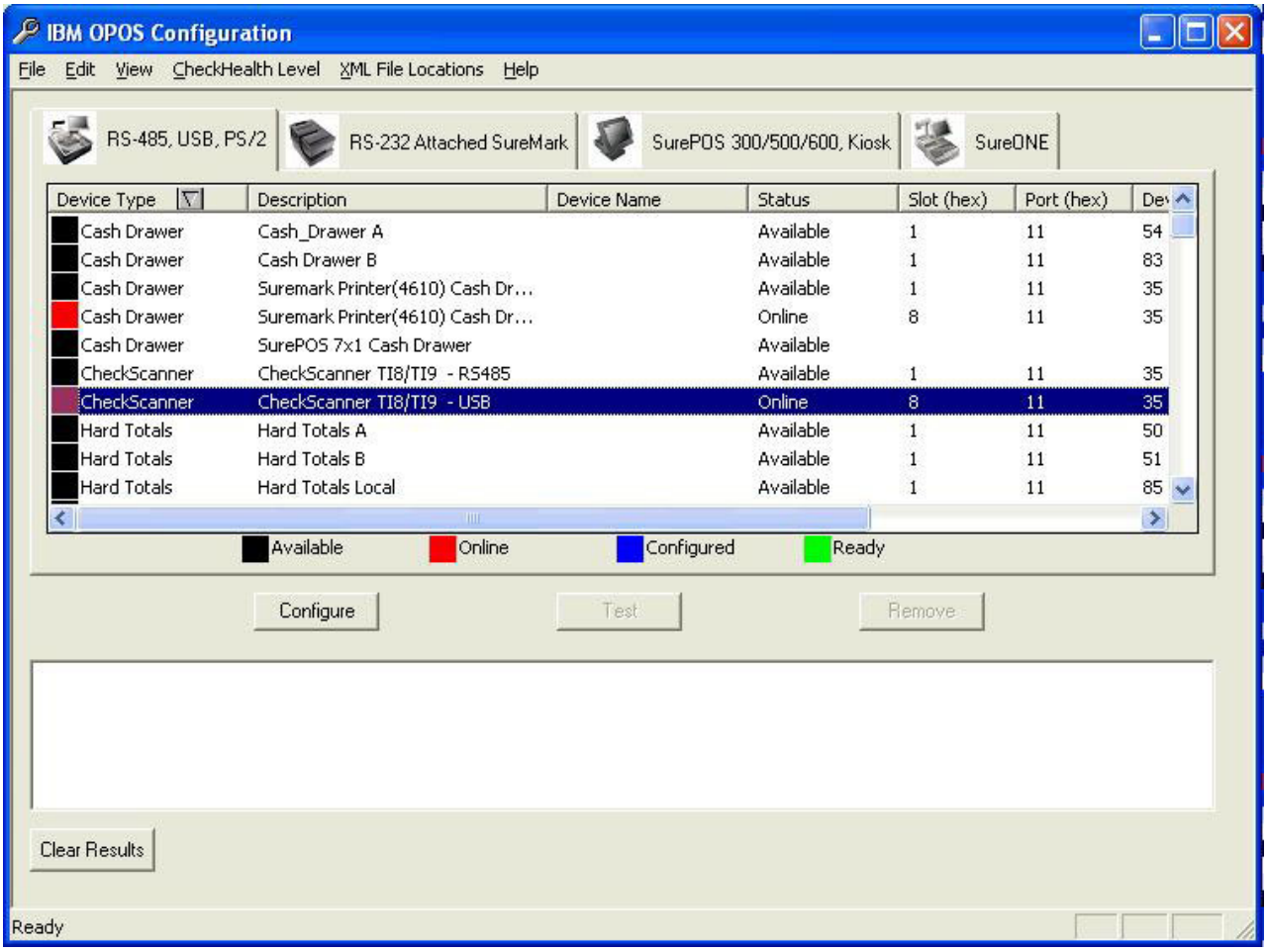


Figure 16. USB CheckScanner configuration panel

Figure 17 on page 49 shows a EIA-232 (RS-232) CheckScanner selected.

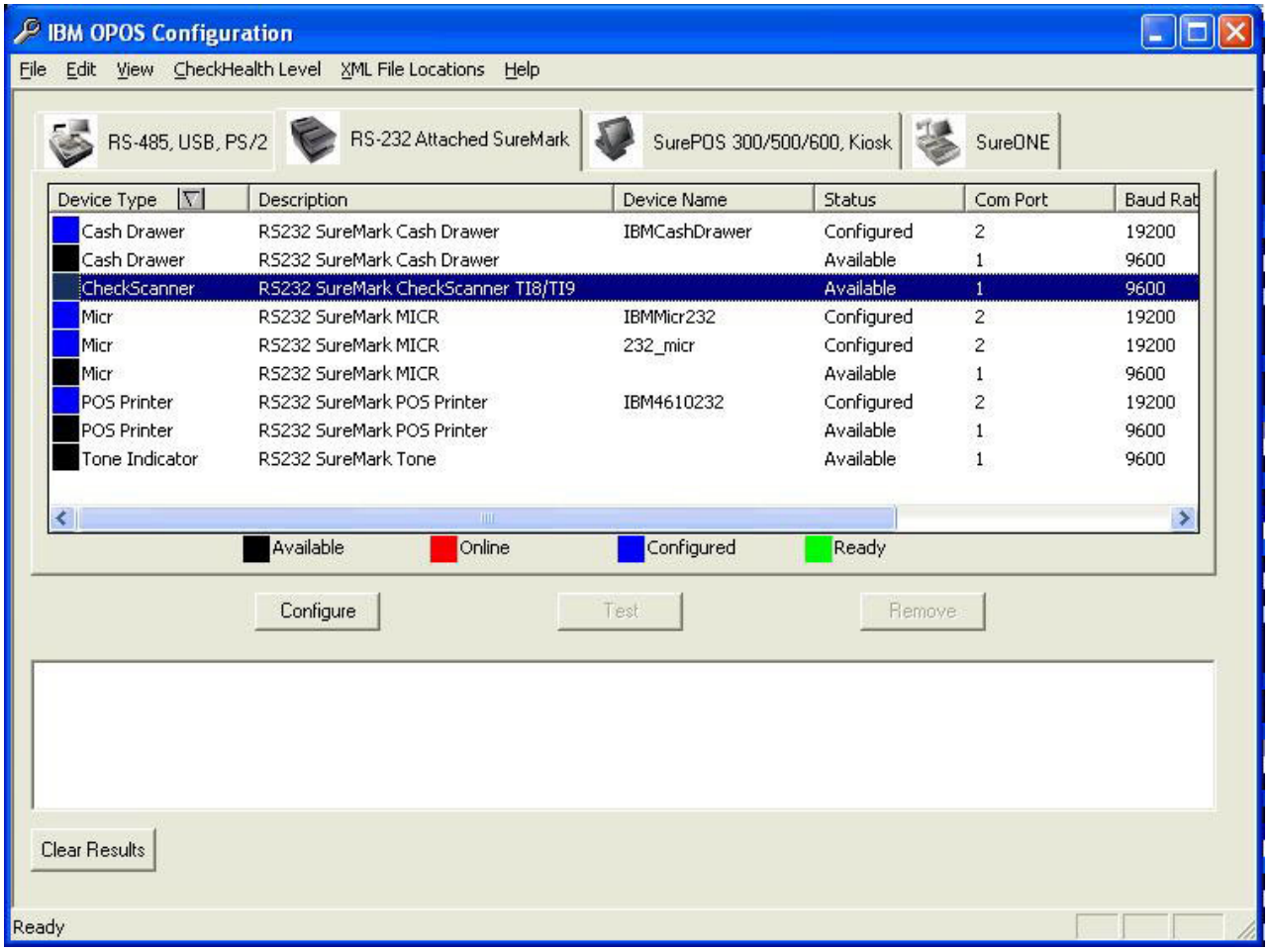


Figure 17. EIA-232 CheckScanner configuration panel

For testing the CheckScanner, click **Test**. The result should show `CheckHealth Test Successful`.

Chapter 4. Device-specific notes

This chapter provides distinctive characteristics about some device configurations.

JavaPOS

This section provides device-specific notes for JavaPOS.

CashDrawer

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

The 4610 Printer-attached Cash Drawer can support regular IBM cash drawers and OEM cash drawers.

- 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, set the signalsReversed property to *true*. For example:

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

Note: The signalsReversed property is located in the JposEntry for 4610 CashDrawer in jpos.xml.<os> Sample.

EIA-232 only:

If the cash drawer is connected to an EIA-232 bus, jpos.xml must contain the following two entries:

- An entry for the IBM 4610 Printer
- An entry for the 4610 CashDrawer for the EIA-232 bus

The *deviceNumber* 0 through 3 are reserved for the CashDrawer attached to the system port.

Loading the CashDrawer device driver for Linux

Add the following into /opt/ibm/javapos/etc/machine.conf

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

The keyword represents a search of the lines that follow to determine if the model of hardware found matches what is in the list. If the model found is listed, then the appropriate driver is installed. If more models are present than found in the list, then add a line for each additional model, and increment the `<number of entries>` value. This value indicates how many lines should be searched for the machine type that was found to be present. For example, the cash drawer driver should look like this:

```
CD 1
4810321
```

CheckScanner

Five directIO commands are supported for the CheckScanner device. The syntax is as follows:

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

When you are using a 4610 Printer-attached CheckScanner with EIA-232 bus (CheckScanner EIA-232 4610 TI8/TI9), you must correctly configure a jpos entry for both the printer and the check scanner. Otherwise the CheckScanner will not be accessible.

CHK_DIO_SCANNER_CALIBRATION_CMD

Command used to initiate calibrate routine for Scanner in Models TI8 and TI9 printers.

Table 15. CHK_DIO_SCANNER_CALIBRATION_CMD information

Command	DirectIO.CHK_DIO_SCANNER_CALIBRATION_CMD
Data	Any value. A null value is accepted.
Obj	Any value. A null value is accepted.

Remarks

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

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

Table 16. UposException information

Value	Meaning
E_ILLEGAL	The following error has occurred: The device is not in check insertion mode.

CHK_DIO_PRINT_SCANNED_IMAGE_CMD

Print the image stored in memory or resident in the work area memory.

Will print the specified scanned image to the thermal station

Table 17. CHK_DIO_PRINT_SCANNED_IMAGE_CMD information

command	DirectIO.CHK_DIO_PRINT_SCANNED_IMAGE_CMD
data[0]	DirectIO.CHK_DIO_MEMORY_IMG prints the most recently scanned image. CHK_LOCATE_BY_FILEINDEX prints the image file using the **FileIndex** property.
data[1]	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]	Percentage to scale image in x direction. (A value of 100 or 0 results in no scaling.)
data[3]	Percentage to scale image in y direction. (A value of 100 or 0 results in no scaling.)

Table 17. CHK_DIO_PRINT_SCANNED_IMAGE_CMD information (continued)

data[4]	If the value is DirectIO.CHK_DIO_ROTATE_90, the image is rotated counter clockwise 90%. If the value is DirectIO.CHK_DIO_NO_ROTATE, the image is not rotated.
obj	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.

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

Table 18. UposException information

Value	Meaning
E_ILLEGAL	The following error has occurred: • Some parameter values are not correct. • The image data file could not be located due to an invalid value stored in either the FileIndex or ImageTagData properties that was being used with data[0] value. • Resident images in the work area memory with CheckScannerConst.CHK_IF_JPEG format.

see also

com.ibm.jpos.services.DirectIO for proprietary constants

CHK_DIO_SET_CHECK_SIDE_CMD

Sets the side of the check to scan. Assumes the check is inserted face down.

Table 19. CHK_DIO_SET_CHECK_SIDE_CMD information

Command	DirectIO.CHK_DIO_SET_CHECK_SIDE_CMD
Data[0]	• 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	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 CHK_DIO_GET_CHECK_SIDE_CMD or data[0] is

CHK_DIO_SIDE_OPPOSITE, the side of the document is changed and the document is fed. If data[0] is CHK_DIO_GET_CHECK_SIDE_CMD, nothing occurs and the method returns.

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

Table 20. ErrorCode property for CHK_DIO_SET_CHECK_SIDE_CMD information

Value	Meaning
E_ILLEGAL	The following error has occurred: Some parameter values are not correct.

CHK_DIO_GET_CHECK_SIDE_CMD

Returns the side of the check to be scanned. Assumes the check is inserted face down. After calling this DirectIO, data[0] will be populated with one of the values in the following table:

Table 21. CHK_DIO_GET_CHECK_SIDE_CMD information

Command	DirectIO.CHK_DIO_GET_CHECK_SIDE_CMD
Data[]	• 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	Any value. A null value is accepted.

Remarks

If the check is not inserted, the value of the property is DirectIO.CHK_DIO_SIDE_UNKNOWN. This property value may be changed when the DirectIO.CHK_DIO_SET_CHECK_SIDE_CMD method is executed. When a check is inserted the value returned is DirectIO.CHK_DIO_SIDE1. However, after beginInsertion/endInsertion methods for POSPrinter processing are performed, the value returned is not limited to DirectIO.CHK_DIO_SIDE1. In this case, the value returned indicates the current side selected of the check scanner.

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

Table 22. UposException information

Value	Meaning
E_ILLEGAL	The following error has occurred: Some of the parameters are not correct.

CHK_DIO_SCANNER_IMAGE_QUALITY_CMD

Command used to 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] will be populated with a value of 0 or 1.

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

Table 23. CHK_DIO_SCANNER_IMAGE_QUALITY_CMD information

Command	**DirectIO.CHK_DIO_SCANNER_IMAGE_QUALITY_CMD**
Data[]	An array with at least one element
Obj	Any value. A null value is accepted.

Remarks

This directIO is supported only by the 4610 TI9 model.

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

Table 24. UposException information

Value	**Meaning**
E_ILLEGAL	The following error has occurred: Some of the parameter values are not correct.

FiscalPrinter

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. This implementation is compatible with previous (legacy) implementations, with the following exceptions:

- Device Category is changed to FiscalPrinter
- New events: FISCAL_RAW_STATUS and FISCAL_END_STATUS

FISCAL_INFORMATION Subcommand

This subcommand is used to obtain information about the fiscal device. It returns valid values after the first FISCAL_IPL_END_STATUS event (after IPL). To perform this operation, use the directIO() method with the parameters shown in Table 25.

Table 25. Parameters for FISCAL_INFORMATION subcommand

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_ID
int [] data	com.ibm.jpos.services.DirectIO.FISCAL_INFORMATION
Object	com.ibm.jpos.services.FiscalInformation Object

Example:

```
public void readFiscalInformation()
{
   FiscalInformation fi=new FiscalInformation();
   int[] cmd = { DirectIO.FISCAL_INFORMATION};
   try
   {
       //sends the command
       printer.directIO(DirectIO.FISCAL_ID, cmd, fi);
       //shows fiscal information
       System.out.println("Fiscal Country: "+fi.getCountry());
       System.out.println("Fiscal Power Interrupted: "+
            fi.getFiscalPowerInterrupted());
       System.out.println("Fiscal Version: "+fi.getVersion());
    }
    catch (JposException e1)
    {
       //handle the exception;
    }
}
```

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 www.ibm.com/solutions/retail/store .)

FISCAL_READ subcommand

This subcommand is used to read data from the fiscal device. It should be issued after receiving a FISCAL_DATA_AVAIL directIOEvent from the fiscal device. To perform this operation, use the directIO() method with the parameters shown in Table 26.

Table 26. Parameters for FISCAL_READ subcommand

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_ID
int [] data	com.ibm.jpos.services.DirectIO.FISCAL_READ
Object	A byte [] to be filled with data.

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_DATA_AVAIL:
      System.out.println("Fiscal data available");
      //as fiscal data is available, we do the reading.
      int len=((Integer) (arg0.getObject()) ).intValue();
      int[] cmd = { DirectIO.FISCAL_READ};
      byte [] buf=new byte[len];
      try
      {
         printer.directIO(DirectIO.FISCAL_ID, cmd, buf);
         //here we can do something with the
         //data in buf (data that we have just read).
         System.out.println("Fiscal Read: "+buf);
      }
      catch (JposException e1)
      {
         //handle the exception;
      }
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

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 subcommand

This subcommand is used to write data to the fiscal device. To perform this operation, use the directIO() method with the parameters shown in Table 27.

Table 27. Parameters for FISCAL_WRITE subcommand

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_ID
int [] data	com.ibm.jpos.services.DirectIO.FISCAL_WRITE
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'}.

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
    //command that prints a memory dump.
    Byte [] obj={ 0xFF, 0x10,
          '0','0','0','0','0',
          '0','0','0','0','F'};
   int[] cmd = { DirectIO.FISCAL_WRITE };
   try
   {
      printer.directIO(DirectIO.FISCAL_ID, cmd, obj);
   }
   catch (JposException e1)
   {
      //handle the exception;
   }
}
```

DirectIO is a synchronous method and will return successfully when the data is written to the serial port; 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 command is ignored. This means that there is no buffering of FISCAL_WRITE subcommands.

FISCAL_Notify subcommand

This subcommand is used to set fiscal notification On or Off. If fiscal notification is On, every fiscal command returns a status event to the application. The following values have been defined:

- com.ibm.jpos.services.DirectIO.FISCAL_NOTIFY_ON
- com.ibm.jpos.services.DirectIO.FISCAL_NOTIFY_OFF

To perform this operation, use the directIO() method with the parameters shown in Table 28.

Table 28. Parameters for FISCAL_NOTIFY subcommand

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_ID
int [] data	com.ibm.jpos.services.DirectIO.FISCAL_NOTIFY
Object	An Integer object with one of the following values: DirectIO.FISCAL_NOTIFY_ON or DirectIO.FISCAL_NOTIFY_OFF

Example:

```
public void turnOnNotify()
{
   //here we will turn on notify
   int[] cmd = { DirectIO.FISCAL_NOTIFY};
   Integer fiscalNotify=new Integer(DirectIO.FISCAL_NOTIFY_ON);
   try
   {
      printer.directIO(DirectIO.FISCAL_ID, cmd, fiscalNotify);
   }
   catch (JposException e1)
   {
      //handle the exception
   }
}
```

FISCAL_ERROR event

This event indicates that a fiscal error has occurred. The directIO parameters are shown in Table 29.

Table 29. Parameters for the FISCAL_ERROR event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_ERROR
int Data	Cause from mp1 of the POSS for Windows fiscal error event. (Fiscal Printer Command Return Code)
Object	Integer object that indicates the mp2 error code as in the POS Subsystem (see Table 30). The Integer value is the result of an OR operation of the current error codes (printer status).

Table 30. mp2 error codes for FISCAL_ERROR event

mp2 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'

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_ERROR:
      System.out.println("Fiscal Error");
      System.out.println("mp1: "+arg0.getData());
      System.out.println("mp2: "+( (Integer) (arg0.getObject()) ).intValue());
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

FISCAL_STATUS event

This event indicates that the last fiscal command completed successfully. Fiscal notification must be On to receive these events. The directIO parameters are shown in Table 31.

Table 31. Parameters for the FISCAL_STATUS event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_STATUS
int Data	Always 0
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 will return the value X'FF' (the first byte indicating the command.) See the Fiscal Printer Hardware Supplements available on the Web at www.ibm.com/solutions/retail/store for more information.

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_STATUS:
      System.out.println("Fiscal Status");
      System.out.println("Last Command: "+
      ( (Integer) (arg0.getObject()) ).intValue());
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

FISCAL_DATA_AVAIL event

This event indicates that data is available to be read from the fiscal device. The directIO parameters are shown in Table 32.

Table 32. Parameters for the FISCAL_DATA_AVAIL event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_DATA_AVAIL
int Data	Always 0
Object	Integer object that indicates the size in bytes of the data to be read (only the electronic response length, non-Fiscal Printer Status and Fiscal Base Status are ignored).

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_DATA_AVAIL:
      System.out.println("Fiscal Data Available");
      System.out.println("Length to read: "+
      ( (Integer) (arg0.getObject()) ).intValue());
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

FISCAL_RAW_STATUS event

This DirectIO event is fired each time any status is received. The directIO parameters are shown in Table 33.

Table 33. Parameters for the FISCAL_RAW_STATUS event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_RAW_STATUS
int Data	Always 0
Object	Byte [] object containing the raw status data

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 www.ibm.com/solutions/retail/store.

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_RAW_STATUS:
      System.out.println("Fiscal Raw Status");
      System.out.println("Raw status: "+(byte[]) arg0.getObject());
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

FISCAL_IPL_END_STATUS event

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. The FISCAL_IPL_END_STATUS event informs the application that the IPL process is complete and that the application can start sending commands to the Fiscal Printer.

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

The directIO parameters are shown in Table 34.

Table 34. Parameters for the FISCAL_IPL_END_STATUS event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.FISCAL_IPL_END_STATUS
int Data	Always 0
Object	Always null

Example:

```
public void directIOOccurred(DirectIOEvent arg0)
{
   switch(arg0.getEventNumber())
   {
      case DirectIO.FISCAL_IPL_END_STATUS:
      System.out.println("Fiscal IPL End");
      System.out.println("IPL process finished.  Printer Ready.");
      break;
      default:
      //in this example we do nothing.
      break;
   }
}
```

LineDisplay

Four 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}
```

DirectIO.SELECT_PLUVFDIIAPA_DISPLAY_TEXT_MODE

Table 35. Parameters for the LineDisplay event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.SELECT_PLUVFDIIAPA_DISPLAY_TEXT_MODE
int Data [0]	0 = display text in normal mode, 1 = display text up-set mode (rotated 180 degrees)
Object	Always null

Remarks

This switch position is typically used for PLU keyboard display units. The display panel is used as either a PLU operator display or a PLU consumer display. When used as a consumer display, the screen image must be up-set (rotated 180 degrees) for the consumer.

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

Table 36. ErrorCode property for SELECT_PLUVFDIIAPA_DISPLAY_TEXT_MODE

Value	Meaning
E_ILLEGAL	The following error has occurred: • -cmd name / value is not known or is not valid. • -Some of the parameter values are not correct.

See also:

com.ibm.jpos.services.DirectIO for proprietary constants

DirectIO.SELECT_APA_VFD_CURSOR_TYPE

Table 37. Parameters for the LineDisplay event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.SELECT_APA_VFD_CURSOR_TYPE
int Data [0]	4 = DISP_CT_UNDERLINE, 1 = DISP_CT_NONE
Object	Always null

Remarks

This is the cursor type for the current window. Changing the ScreenMode property also changes the way the CursorType is displayed. DISP_CT_UNDERLINE can be set only if the current ScreenMode property or current number of rows is "4x24" or "5x20". DISP_CT_NONE can be set only if current ScreenMode property or current number of rows is "2x20".

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

Table 38. ErrorCode property for SELECT_APA_VFD_CURSOR_TYPE

Value	Meaning
E_ILLEGAL	The following error has occurred: • -cmd name / value is not known or is not valid. • -Some of the parameter values are not correct.

See also:
com.ibm.jpos.services.DirectIO for proprietary constants

DirectIO.SELECT_APA_NUMBER_OF_ROWS

This command applies for EIA-232 APA (all points addressable) character graphics displays (160x40 APA VFD).

This command can also be used with ScreenMode property.

Table 39. Parameters for the LineDisplay event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.SELECT_APA_NUMBER_OF_ROWS
int Data [0]	2 = "2x20" 2 rows by 20 columns, 4 = "4x20" 4 rows by 20 columns, 5 = 5x20 5 rows by 20 columns
Object	Always null

Remarks
This is the screen mode value of the device. Changing the ScreenMode property also changes the way the DeviceColumns, DeviceRows, GlyphHeight, GlyphWidth, Rows, and Columns properties to the new screen size. Changing the current screen mode can be done only if the current language permits it.

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

Table 40. ErrorCode property for SELECT_APA_VFD_CURSOR_TYPE

Value	Meaning
E_ILLEGAL	The following error has occurred: • -cmd name / value is not known or is not valid. • -Some of the parameter values are not correct.

See also:
com.ibm.jpos.services.DirectIO for proprietary constants

DirectIO.SELECT_ROTATE_TEXT_180

This command applies for USB/HID APA Vacuum Fluorescent Display (VFD).

Table 41. Parameters for the LineDisplay event

Parameter	Value
int command	com.ibm.jpos.services.DirectIO.SELECT_ROTATE_TEXT_180

Table 41. Parameters for the LineDisplay event (continued)

Parameter	Value
int Data [0]	0 = display text in normal mode, 1 = display text in up-set mode (180 degrees rotated)
Object	Always null

Remarks

The value of the converse position of the dip switch is obtained at the POR initialize process only, and the display mode changes to normal or conversed. Bit 7 of the Write mode command can change this mode.

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

Table 42. ErrorCode property for SELECT_APA_VFD_CURSOR_TYPE

Value	Meaning
E_ILLEGAL	The following error has occurred: • -cmd name / value is not known or is not valid. • -Some of the parameter values are not correct.

See also:

com.ibm.jpos.services.DirectIO for proprietary constants

Table 43. Quick reference for LineDisplay DirectIO command

LineDisplay	Command
Character Graphic (C/G) RS485	DirectIO.SELECT_PLUVFDIIAP A_DISPLAY_TEXT_MODE or DirectIO.SELECT_ROTATE_TEXT_180 *data* [0] 0 = display text in normal mode 1 = display text up-set mode (rotated 180 degrees) *Obj = null*
Character Graphic (C/G) EIA232 APA	DirectIO.SELECT_APA_VFD_CURSOR_TYPE *data* [0] 4 = DISP_CT_UNDERLINE. 0 = DISP_CT_NONE *Obj = null* Remarks: Holds the cursor type for the current window. Changing the ScreenMode property also changes the way the CursorType is displayed. DISP_CT_UNDERLINE can be set only if the current ScreenMode property or current number of rows is 4X20 or 5X20. DISP_CT_NONE can be set only if the current ScreenMode property or current number of rows is 2X20.
	DirectIO.SELECT_APA_NUMBER_OF_ROWS *data [0]* 2 = 2X20 (2 rows by 20 columns) 4 = 4X20 (4 rows by 20 columns) 5 = 5X20 (5 rows by 20 columns) *Obj = null*

Keylock

This section describes the available keylocks.

Position count configuration

A user-configurable mechanism is provided to define the number of key lock positions. This is accomplished by adding the following property to the keylock's JPOSEntry:

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

Some identified devices that might not set the correct number of keylock positions are:

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

Two-position keylock

The keylock for the NANPOS,50–KEY, 133–KEY, and 4820 Point of Sale Keyboards has two positions and reports these as LOCK_KP_NORM and LOCK_KP_SUPR.

Note: LOCK_KP_LOCK is not available.

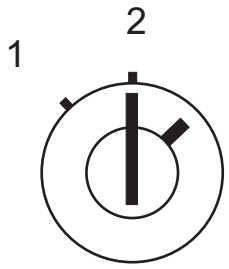


Figure 18. Two-position keylock

Key position	Description	Key position value
1	Operation	LOCK_KP_NORM
2	Manager	LOCK_KP_SUPR

Three-position keylock

The keylock for the SureOne has three positions

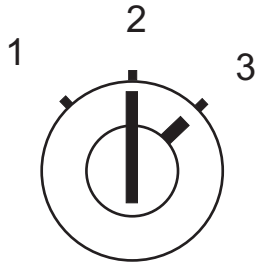


Figure 19. Three-position keylock

Key position	Description	Key position value
1	Inactive	LOCK_KP_LOCK
2	Operation	LOCK_KP_NORM
3	Manager	LOCK_KP_SUPR

Four-position keylock

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

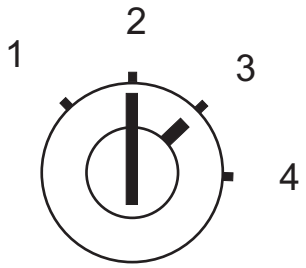


Figure 20. Four–position keylock

Key position	Description	Key position value
1	System	4
2	Inactive	LOCK_KP_LOCK
3	Operation	LOCK_KP_NORM
4	Manager	LOCK_KP_SUPR

Five-position keylock

The keylock for the IBM 4674 keyboard has five positions.

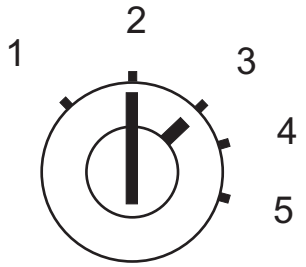


Figure 21. Five-position keylock

Key position	Description	Key position value
1	PC Mode	5
2	Inactive	LOCK_KP_LOCK
3	Operation	LOCK_KP_NORM
4	Manager	LOCK_KP_SUPR
5	System	4

Six-position keylock

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

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

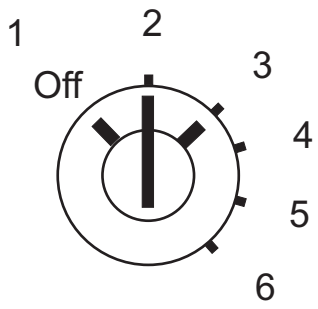


Figure 22. Six-position keylock

Key position	Description	Key position value
1	Inactive	LOCK_KP_LOCK
2	Operation	LOCK_KP_NORM
3	User-defined	5
4	User-defined	6
5	Manager	LOCK_KP_SUPR
6	System	4

MICR

This section describes Magnetic Ink Character Recognition (MICR) device-specific notes

4610 POSPrinter attached MICR (EIA-232 bus only)

When you are using a 4610 Printer-attached MICR with EIA-232 (formerly known as RS-232) bus, you must configure a jpos entry for both the Printer and the MICR. If you don't have both, the MICR won't populate.

MICR Exception Tables (applies for all platforms and buses)

The format of the data read from the 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.

Given this, only On Us and Auxiliary On Us fields need interpretation rules.

As a result, the Exception Table consists only of entries that have the Transit Field-On-Us Field interpretation rule pair. 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 separated file. Details on how to use this file are given in the description of the first format. Formats two (2) and three (3) can be put directly into the MICR jpos entry. It is worth noting that this file can also contain definitions for the last two formats.

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 may be pointed by a property in the jpos entry inside 'jpos.xml':

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

If the property is omitted then the filename “ibmmicrexception.xml” is searched throughout the classpath. 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. See the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MICRExceptionTable SYSTEM "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>
```

The following considerations must be taken:

- There can be as many `ExceptionEntry` or `prop` elements as needed and any `ExceptionEntry` element 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`. If omitted it defaults to `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 jpos entry. 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 you find them here is the order you expect the fields to appear in the On-Us field. `OnUsFieldParsingRule` is required.
- `SerialNumberLength` and `AccountNumberLength` can be in either order, and `SkipCharacterLength` can be at the beginning or between `SerialNumberLength` and `AccountNumberLength`.
- There should be exactly one occurrence of `AccountNumberLength` (required).
- `SerialNumberLength` and `SkipCharacterLength` are optional.
- All 'value' attributes are required. `TransitField` attribute from `ExceptionEntry` element is also required.

- `CheckType` attribute from `ExceptionEntry` must be either *Personal* or *Business* and it is optional. If omitted, it defaults to *Personal*.
- Depending on the company's format, the serial number may appear in the Auxiliary On-Us field, `AuxOnUsFieldParsingRule` will hold 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 *value* is parsed as shown in Table 44.

Table 44. MICR exception table parsing

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 JPOS.xml. The format is:

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

where each byte represents the following:

P/B (byte 1)
Type of check (business or personal). Not used for parsing. Used to set the check type property, which is usually set to UNKNOWN.

TRANSIT (byte 2-10)
Bank transit number. Used as the key to determine if special processing is required

SPC (byte 11)
If SPC is set to D, all spaces are removed from the On-Us field for processing; otherwise, the spaces are not removed.

FIELD (byte 12)
Indicates which subfield in the On-Us field contains the account number.

SKIP (byte 13)
Index of the character that contains the starting point of the account number in the On-Us field. (The index count starts at 0.)

SIZE (byte 14)
Length of the serial number

ACC (byte 15)
Alignment of the account number:

- L aligns on the left of the field.
- R aligns on the right of the field.

JPOS Entry Settings: **com.ibm.jpos.sdi.config.MICR.stripAccountDashes**

The default value for this property is *false* and no change is made to the Account number. When it is set to *true*, the dashes are removed from the Account Number, so a value of ″1234-67-9″ becomes ″1234679″.

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

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

The default value for this property is *false* and no change is made to the Transit number. When it is set to *true*, the dashes are removed from the Transit Number, so a value of ″7777-77-″ becomes ″777777″.

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

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

The default value for this property is *false* and no change is made to the Transit number. When it is set to *true*, the dashes are changed to spaces, so a value of ″12-4-66-7″ will become ’12 4 66 7″.

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

MotionSensor

This section describes how to load a device driver in Linux.

Add the following into /opt/ibm/javapos/etc/machine.conf

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

The keyword represents a search of the lines that follow to determine if the model of hardware matches what is in the list. If the model found is listed in the searched lines, then the appropriate driver is installed. If more models are present than found in the list, then add a line for each additional model, and increment the <number of entries> value. This value indicates how many lines should be searched for the machine type that is found to be present. For example, the motion sensor driver for a 4838 should look like this:

```
Motion 2
4838132
4838135
```

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

MSR

The following directIO is supported on IBM 4685-K02 MSR:

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

command
DirectIO.MSR_WRITE_DATA_CMD

data[0]
Write wait timeout value in milliseconds

obj String representation with the data to be written

Remarks

Command used to write data into the MSR card. Data can only be written to track JISII.

The wait timeout can be set to JposConst.JPOS_FOREVER and the command will have no timeout to return. The maximum wait timeout is 25,000 milliseconds (25 seconds).

The maximum length for String representation of the data Maximum is 69. The rest of the data is ignored if the data is greater.

A card with track JISII 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 (see “Errors”).

Errors

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

Table 45. UposException

Value	Meaning
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.

SureOne

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.

IBM 4836/4838 MSR

enableOnlineWatcher property:

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

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

EIA-232 MSR

To set an ISO MSR, the property msrType can be set to either "1" or "ISO".

```
<prop name="com.ibm.posj.bus.rs232.msrId" type="String" value="ISO"/>
or
<prop name="com.ibm.posj.bus.rs232.msrId" type="String" value="1"/>
```

POSKeyboard

Remapping scan codes

Per the UnifiedPOS specification, a user-configurable mechanism is provided to define the keycodes that the application receives for the actual scan codes obtained from the keyboard. This is accomplished by identifying in the keyboard's JPOSEntry (through the KeyTranslationFile property) the name of a Keyboard Definition Table text file that contains the translation information. For example:

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

The value attribute should indicate the full path name of the file with the keyboard definition table. A sample file called keys.dat.Sample is included with the installation. This can be useful as a starting point for adding settings for the keyboard. The location for keys.dat.Sample is:

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

Keyboard definition table (\keys.dat):

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 Escape, [, and A.)

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

```
[302 303] 141
```

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

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

Note that in this case the virtual scancode sequence 312-320-322 will be generated at once only when the actual keyboard codes sequence completes. This means that the single 141 code will not trigger any scancode.

To make it easier to write a keyboard definition table, some additional ways to encode characters are allowed. Here are the character-encoding rules:

1. Any single character, except space, tab, and # stands for itself. Thus, the example sequence could have been coded as: `301 01 [ A`
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 could have been 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 would be taken as an ASCII character under Rule 1. Thus, 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, =, ...) need not 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`, 100 is the scan code for `b`, the entries in the `keys.dat` file are:

   ```
   400 99 c
   ```

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

   – Type the 'a' key (scan code 99). The 'a' key will not return scan codes until scan code completes the sequence.
   – Then type 'b'. Then, 'a' will 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.

Note: 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 user-configurable mechanism provides a way to return a single scan code when a double-sized lens is used over two single key buttons. The following steps describe on how to create a double key.

1. Add double definitions to a text file (such as keys.dat).
2. Add the following property to JposEntry for the keyboard in jpos.xml file:

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

The double key definition format is as follows:

```
0xFF <expected scan code> <scancode for 1st key> <scancode for 2nd key>
```

For example:

```
0xFF 0x9E 0x9E 0x9D
```

where:

0xFF defines double sequence

0x9E expected scancode

0x9E scancode for the first key

0x9D scancode for the second key

0x9E scancode is returned to the application.

Keyboard Click Configuration

A user-configurable mechanism is provided to modify the default configuration. This is accomplished by identifying in the keyboard's JPOSEntry the following definitions:

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

Click : Each keyboard will provide a high or low volume key click through the speaker. Possible Values:

0x00 = Click OFF - Default Value

0x01 = Click Soft

0x02 = Click Loud

Keyboard typematic delay rate

- `<prop name="com.ibm.jpos.sdi.config.POSKeyboard.Typematic" type="Boolean" value="true"/>`
- `<prop name="com.ibm.jpos.sdi.config.POSKeyboard.TypematicDelay" type="Byte" value="0x01"/>`
- `<prop name="com.ibm.jpos.sdi.config.POSKeyboard.TypematicRate" type="Byte" value="0x14"/>`

  Most of the keys on all keyboards are optionally typematic keys.

  Through this JposEntry, you can enable or disable the typematic function for the entire keyboard.

  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 a key is pressed, 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 as 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.

  – Typematic values:
    true = enabled (default value)
    false = disabled
  – TypematicDelay values (msec. +/- 20%):
    0x00 = Delay 250
    0x01 = Delay 500 (default value)
    0x02 = Delay 750
    0x03 = Delay 1000
  – TypematicRate values (make codes/sec. +/- 20%):
    0x00 = Rate 2.0
    0x01 = Rate 2.1
    0x02 = Rate 2.3
    0x03 = Rate 2.5
    0x04 = Rate 2.7
    0x05 = Rate 3.0
    0x06 = Rate 3.3
    0x07 = Rate 3.7
    0x08 = Rate 4.0
    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 (default value)
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

Keyboard scanning

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

KbdScanning : When the keyboard is enabled for key detection, the keyboard generates make and break scan codes. Possible values:
true = enabled (default value)
false = disabled

Keyboard fat finger timeout

```
<prop name="com.ibm.jpos.sdi.config.POSKeyboard.FatFingerTimeout"
type="Byte" value="03"/>
```

Fat-Finger: 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. It the two keys were pressed within a certain interval, a "fat-finger" situation is assumed, and a status indicating this occurrence accompanies the key scan codes to the terminal. Possible values:

- 0x00 = 00 msec (this disables fat-finger detection)
- 0x01 = 10 msec
- 0x02 = 20 msec
- 0x03 = 30 msec
- 0x04 = 40 msec

Note: The TypematicDelay, TypematicRate, and FatFingerTimeout JposEnbtries are not supported by way of the PS/2(55) keyboard port and system attached USB NANPOS keyboard.

Extended keys support (PS/2 Keyboard)

The extended scan code format was implemented 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').

The ExtendedKeyMapping property, when *true*, retains the extended key mapping format, but modifies it as well. The X'e0' scan code is replaced by X'01' without affecting the subsequent scan code. For example, the hardware scan code X'e0' X'23' is returned to the application as X'123'.

If the ExtendedKeyMapping property is set to *false*, the extended scan code format is not supported. Only the one-byte scan code is returned to the application (X'0023').

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

ExtendedKeyMapping Values:

- true = enabled (default value)
- false = disabled

USB System Attached Keyboard on Linux

There is a limitation for USB alphanumeric keyboards functioning as a system keyboard: 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.

Keyboard DirectIO Commands

There are three DirectIO commands for the keyboard:

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

Table 46. DirectIO.GET_KBD_LIGHT_STATUS_CMD

command	**DirectIO.GET_KBD_LIGHT_STATUS_CMD**	
data	An `int` array with at least one element	
object	Any value. A null value is accepted	
Description	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	
The bit representation is:		
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

Table 47. DirectIO.TURN_KBD_LIGHT_ON_CMD

command	**DirectIO.TURN_KBD_LIGHT_ON_CMD**
data[0]	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
object	Any value. A null value is accepted
Description	This command illuminates the indicator specified on the data[0] parameter.

Table 48. DirectIO.TURN_KBD_LIGHT_OFF_CMD

command	**DirectIO.TURN_KBD_LIGHT_OFF_CMD**
data[0]	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
object	Any value. A null value is accepted
Description	This command turns off the indicator specified on the data[0] parameter. When the `DirectIO.KBD_LIGHTS_OFF` data parameter is used for `DirectIO.TURN_KBD_LIGHT_OFF_CMD` command then all the indicators are turned off.

Errors: A UposException might be thrown when this method is invoked. See on page 91. Some possible values of the exception's *ErrorCode* property are:

E_ILLEGAL
The following error has occurred:

Some of the parameter values are not correct.

See also com.ibm.jpos.services.DirectIO for proprietary constants.

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 www.ibm.com.
2. Type **CANPOS utility** in the search field.
3. Click **IBM RSS Support - Other**.
4. Scroll down to **Compact ANPOS Keyboard**. There are three versions available supporting different operation systems. Based on your preferred system, you can download the appropriate utility.
5. Unzip the package and use the utility to generate the configuration file.
6. 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.
7. Reboot your POS system and the configuration file will be 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.

Note: The CANPOS keyboard does not support keyboard directIO.

Installing USB System Attached Keyboard on Windows XP

The proper device driver is automatically installed for the USB System Attached Keyboard on Windows XP. If you select **IBM Alphanumeric Point of Sale Keyboard** and indicate that it is attached to a USB port, the system reboots. 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.

POSPrinter

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 properties *LineHeight 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.

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

Some of the first 31 chars of the ASCII code are non-printable characters, the POSPrinter handles those as strings and prints the hexadecimal value of that character in the string. The following table shows the values that the POSPrinter prints for these values.

Table 49. 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
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

Table 49. Handling invalid characters (continued)

Hex Value:	Character:	Also known as:	JavaPOS print:
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.

Line wrapping on the slip station while in sideways mode (PTR_RP_LEFT90)

The hardware cannot determine the width of the paper while in sideways mode; therefore, it is not possible to perform automatic wrapping in this case. Applications should not assume the printer service will provide automatic wrapping before reaching the edge of the paper while in this mode. Setting the SlpLineChars value to a valid number will wrap correctly.

In RotatePrint mode sideways on the slip station, only the following escape sequences are supported:

- Top and bottom log
- Double wide
- Left, center, and right alignment

Bitmap printing

- To use setBitmap and printBitmap in a non-graphics environment with 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.
- Bitmap sizes are limited based on the type of printer being used and the station being printed to. If a TOOBIG error is being received, check the size of the bitmap to make sure it fits within the maximum sizes shown in Table 50.

Table 50. Bitmap printing

Type of printer	Station	Method	Property	Max height	Size
4610 TI 3/4/5	receipt	printBitmap()	rlw	None	None
4610 TI 3/4/5	receipt	setBitmap()	rlw	2040	w*h/8<8K
4610 TI 3/4/5	slip	printBitmap()	slw	None	None
4610 TI 3/4/5	slip	setBitmap()	slw	40	w*h/8<8K

Notes:

1. rlw = RecLineWidth property
2. slw = SlpLineWidth property
3. 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.)

Escape sequence handling

- The Fire Stamp escape sequences are supported on the 4610 printers. They use the bitmap downloaded as bitmap #1 for the stamp.
- Handling of invalid, unrecognized, or illegal value escape sequences are handled 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. ***They are not passed to the printer.*** This is based on the specification's use of the term ″ignored.″ Since ignored is not passed to the printer, 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.

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 will be used, because enabling the feature slows down the printer, whether or not color printing is performed.

Direct IO Functions/Events

The device must be claimed and enabled before invoking the directIO() method.

Check flipping support

While the drivers still support the DirectIO.FLIP_CHECK_ID, the preferred method to flip the check is to use:

```
changePrintSide(  )
```

Font Information Retrieval (IBM 4610 Printers)

Font Information Retrieval is supported using a directIO() call. The GET_FONT_INFO Subcommand is used to obtain information about the fonts in the IBM 4610 Printer. To perform this operation, use the directIO() method as follows:

- Parameters:
  - command: com.ibm.jpos.services.DirectIO.GET_FONT_INFO_ID

  - data: int[1] = com.ibm.jpos.services.DirectIO.GET_FONT_INFO
  - object: com.ibm.jpos.services.sdiibmprinter.PrintGetDirectIOFontInfo (EIA-232 and USB (Linux))
  - com.ibm.jpos.services.ibmprinter.DirectIOFontInfo (RS-485 and USB (Windows))
- Example:

```
int[] data = new int[] { DirectIO.GET_FONT_INFO };
PrintGetDirectIOFontInfo directIOFontInfo
        = new PrintGetDirectIOFontInfo();
printer.directIO(DirectIO.GET_FONT_INFO_ID,
       data,
       directIOFontInfo);
```

Font Download (IBM 4610 and 4689 Printers)

Font Download is supported using a directIO() call and the command line utility, AipFnt46s. There are a total of four code pages in the receipt station while the slip station has two supported code pages.

Note: Downloading to code page 1 of the Impact/Thermal station erases **all** fonts in that particular station.

DirectIO Methods: **DirectIO method - com.ibm.jpos.services.DirectIO. DOWNLOAD_PROP_FONT_ID**

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

Note: Proportional spacing is not supported by the 4689 Printer.

Parameter
Value

Command
DOWNLOAD_PROP_FONT_ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter. DirectIOFontInfo

Description
Used to download a proportional font to the IBM 4610 Printer. DirectIOFontInfo receives 3 arguments when it is created:

1. station
2. codepage
3. filename

- Station int - the station to download the fonts, proportional fonts are supported only in Receipt station.
- Codepage - int - the integer values 1 and 3. Where: 1 represents the first codepage of the Receipt station 3 represents the third codepage of the Receipt station.
- Filename - string - file name of the font file must be in the user's ABSOLUTE Path.

Note: 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.

Example:

```
int[] data = new int[5];
DirectIOFontInfo directIOFontInfo = new DirectIOFontInfo( int station,
                          int codepage,
                          String fileName );
printer.directIO(DirectIO.DOWNLOAD_PROP_FONT_ID, data, directIOFontInfo);
```

After a successful font download (proportional or fixed), use the characterSet property in order retrieve and set the desired code pages for font printing. Table 51 shows the usage of the user-defined character sets.

Table 51. User-defined Character Sets

Character Set	Font Attribute	CR Station	DI Station
101	Fixed	User Defined (UD) Code Page 1	UD CodePage 1
	Proportional	UD Code Page1	N\A
102	Fixed	UD Code Page2	N\A
	Proportional	N\A	N\A
103	Fixed	UD Code Page3	UD Code Page 2
	Proportional	UD Code Page3	N\A
104	Fixed	UD Code Page4	N\A
	Proportional	N\A	N\A

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

Table 52. 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 not allow proportional fonts • Incorrect font number • Incorrect font file type
E_EXIST	Memory sector already has a font, erase the memory sector before downloading a font.
E_NOEXIST	Font file was not found

DirectIO Methods: **DirectIO method - com.ibm.jpos.services.DirectIO. DOWNLOAD_NON_PROP_FONT_ID**

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

Parameter
Value

Command
DOWNLOAD_PROP_NON_FONT_ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter.DirectIOFontInfo. Class that holds the printer station, font number and file name of the font to download.

Description

Used to download a non-proportional font to the IBM 4610 and 4689 printers. DirectIOFontInfo receives 3 arguments when it is created:

1. station
2. codepage
3. filename

- Station int - the station to download the fonts, valid values for 4610 are: POSPrinterConst.PTR_S_RECEIPT, POSPrinterConst.PTR_S_SLIP, this parameter is ignored by 4689 Printer since font is downloaded to both receipt and journal stations.
- Codepage - int - the integer values 1 - 6. Where: 1 - 4 represents the four codepages of the receipt station - 5 - 6 represents the two codepages in the slip station. This parameter is ignored by the 4689 printer.
- Filename - string - file name of the font file must be in the user's ABSOLUTE Path.

Note: Downloading to code page 1 of the Impact/Thermal station erases ALL fonts in that station.

Example:

```
int[] data = new int[5];
DirectIOFontInfo directIOFontInfo = new DirectIOFontInfo( int station,
                          int codepage,
                          String fileName );
printer.directIO(DirectIO.DOWNLOAD_NON_PROP_FONT_ID, data, directIOFontInfo);
```

After a successful font download (proportional or fixed), use the characterSet property to retrieve and set the desired code pages for font printing. Table 53 shows the usage of the different user-defined character sets.

Table 53. User-defined Character Sets

Character Set	Font Attribute	CR Station	DI Station
101	Fixed	User Defined (UD) Code Page 1	UD CodePage 1
	Proportional	UD Code Page1	N\A
102	Fixed	UD Code Page2	N\A
	Proportional	N\A	N\A
103	Fixed	UD Code Page3	UD Code Page 2
	Proportional	UD Code Page3	N\A
104	Fixed	UD Code Page4	N\A
	Proportional	N\A	N\A

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

Table 54. UposException Error Codes

Value	Meaning
E_ILLEGAL	One of following error has occurred: • This printer does not support proportional fonts downloading. • This Printer station does not allow proportional fonts. • Font number is incorrect. • Font file type is incorrect.
E_EXIST	Memory sector already has a font. Erase the memory sector before downloading a font.
E_NOEXIST	Font file was not found.

DirectIO Methods: **DirectIO method - com.ibm.jpos.services.DirectIO. DOWNLOAD_DBCS_FONT_ID**

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

Parameter
Description

Command
DOWNLOAD_DBCS_FONT_ID_ID DirectIO ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter.DBCSFontInfo. Class that holds the filename of the DBCS font to download.

Description
Used to download a double-byte font to the printer. The printer must be double-byte. DBCSFontInfo constructors receives the file name of the font file, font file must be in the user's ABSOLUTE path.

Example:

```
int[] data = new int[5];
DBCSFontInfo dbcsFontInfo = new DBCSFontInfo( String fileName );
printer.directIO(DirectIO.DOWNLOAD_DBCS_FONT_ID, data, dbcsFontInfo);
```

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

Table 55. UposException Error Codes

Value	Meaning
E_ILLEGAL	One of the following errors has occurred: • This printer does not support DBCS fonts downloading. • This Printer station not allow proportional fonts. • Incorrect font file type.
E_NOEXIST	Font file was not found.
E_FAILURE	Error reading from font file.

DirectIO Methods: **DOWNLOAD_PROP_FONT_ID**

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

Parameter
Value

Command
DirectIO.DOWNLOAD_PROP_FONT_ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter.DirectFontInfo. Class that holds the printer station, font number, and filename of the font to download.

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

Table 56. 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 not allow proportional fonts • Incorrect font number • Incorrect font file type
E_EXIST	Memory sector already has a font, erase the memory sector before loading a font.
E_NOEXIST	Font file was not found

DirectIO Methods: **DirectIO method - com.ibm.jpos.services.DOWNLOAD_NON_PROP_FONT_ID**

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

Parameter
Value

Command
DirectIO.DOWNLOAD_NON_PROP_FONT_ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter.DirectFontInfo. Class that holds the printer station, font number, and filename of the font to download.

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

Table 57. 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 not allow non proportional fonts • Incorrect font number • Incorrect font file type
E_EXIST	Memory sector already has a font, erase the memory sector before loading a font.
E_NOEXIST	Font file was not found

DirectIO Methods: **DirectIO method - com.ibm.jpos.services.DOWNLOAD_DBCS_FONT_ID**

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

Parameter
Description

Command
DirectIO.DOWNLOAD_DBCS_FONT_ID

Data *not used*

Obj com.ibm.jpos.services.sdiibmprinter.DBCSFontInfo. Class that holds the filename of the DBCS font to download.

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

Table 58. UposException Error Codes

Value	Meaning
E_ILLEGAL	One of the following errors has occurred: • This printer does not support DBCS fonts downloading • Incorrect font file type
E_NOEXIST	The image file does not exist

Command Line Utility (AipFnt46s): A command utility is provided to download the fonts to the printer. The syntax for the utility is:

```
java com.ibm.jpos.util.font.AipFnt46s [param] [filename] [codepage]
```

Valid values for the variables are:

param

-f: fixed font file

-p: proportional font file

-d: dbcs font file (valid only for DBCS supported printers)

filename

Includes the absolute pathname of the font file (For example: `/opt/ibm/javapos/fonts/FontFile.fTH`)

codepage

1: Receipt station code page one

2: Receipt station code page two

3: Receipt station code page three

4: Receipt station code page four

5: Slip station code page one

6: Slip station code page two

0: DBCS font download only

Firmware Flash Update

The firmware can be updated by automatic update or by the command line utility.

Automatic Update: Follow these steps to update the POSPrinter firmware for Linux:

1. Obtain the appropriate flash file from the Web:
   - aip46mch.hex for 4610 Tx3, Tx4 and Tx6
   - aip46mcd.hex for 4610 TI5 and TI7
   - aip46ti8.hex for TI8 and TI9

   Go to: `www.ibm.com/solutions/retail/store`. Click **Support** and follow the links to JavaPOS POSPrinter support.
2. Create an `rs485` directory as follows:
   - For Linux, go to the directory /opt/ibm/javapos/flash on the target system and create an rs485 directory. This is done as shown in the following example:

     ```
     % cd /opt/ibm/javapos/flash
     % mkdir rs485
     ```
   - For Windows, go to the directory `<root install directory>\IBMJPOS\res\flash` on the target system and create an rs485 directory. This is shown in the following example, where the root install directory is C:\POS.

     ```
     C:\> cd\POS\IBMJPOS\res\flash
     C:\> mkdir rs485
     ```
3. Place the correct flash file (ensure it matches the printer type) into the new directory.
4. Make sure the printer is attached and reboot the system. (The flash will take a few minutes to complete during boot-up.)

Command Line Utility (AipFlash46s): The IBM 4610 printer firmware can be downloaded using a command utility. The syntax for the utility is:

`java com.ibm.jpos.util.flash.AipFlash46s [`*param*`] [`*RS485_flash_filename*`]`

Valid values for the variables are:

param

-e: flash printer EC level

RS485_flash_filename

Includes the absolute pathname of the flash file (For example: `/opt/ibm/javapos/flash/aipmch46.hex`)

Page Mode support

Note: Direct I/O page mode support is deprecated. The preferred method is the page mode support in the UPOS 1.9 specification.

When in PageMode, transactionPrint and rotatePrint are prohibited and will throw E_ILLEGAL. 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.

Direct IO properties: The properties of the PageMode support for IBM printers can be accessed using directIO commands. These commands are handled indirectly. When the properties command is called, the data parameter is set to the supported value for that printer.

PageMode properties:

- AfterOpen (no station):
  - PageModeArea = ""
  - PageModePrintArea = ""
  - PageModeHorizontalPosition = 0

  - PageModeVeritcal Position = 0
  - PageModePrintDirection = 0
  - PageModeStation = 0
- After setting a valid station:
  - PageModeArea = ″576,800″ (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″ (1250 for T*x*8/T*x*9). If it still has the default value, this value is set.
- Setting PrintDirection: Setting this property may change PageModeHorizontalPosition and PageModeVerticalPostion.
- Setting Vertical or Horizontal Position: If the position set is not supported, then is it set to its closest supported value. If you set values outside of PageModePrintArea, the drivers set these values to the size of PageModePrintArea.

Page Mode is a design mode. The vertical and horizontal position properties must be set before sending any print command, otherwise the printed image can be cut off.

Figure 23. Partial image, no position setting

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",-11,-2);
printer.pageModePrint(jpos.POSPrinterConst.PTR_PM_NORMAL);
```

In the next example the printing is correct, with position properties set.

Figure 24. Correct printing

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",-11,-2);
printer.setPageModeHorizontalPosition(0);
printer.setPageModeVerticalPosition(300);
printer.printBarCode( POSPrinterConst.PTR_S_RECEIPT,"01234567", 108,
        100, 200, -1, -13);
printer.pageModePrint(jpos.POSPrinterConst.PTR_PM_NORMAL);
```

For the image to be positioned correctly, set the coordinates to (0, <height of bitmap>).

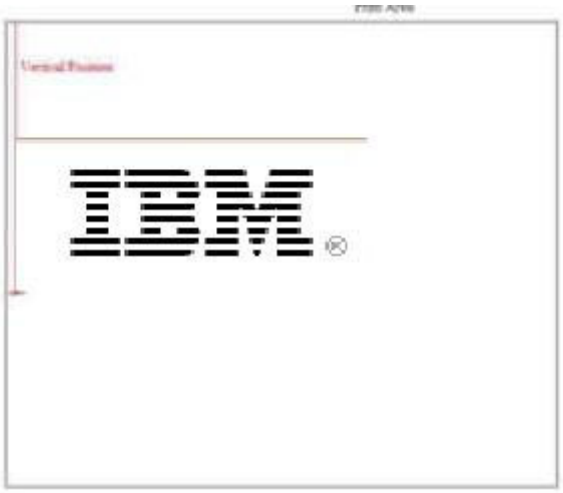


Figure 25. Correct position

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",-11,-2);
printer.pageModePrint(jposPOSPrinterConst.PTR_PM_NORMAL);
```

Printing text: Begin in the next line.

Figure 26. Two lines of text

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);
```

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.

Figure 27. Image follows 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", -11, -2);
printer.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

When you print an image or barcode, the following text, image, or barcode prints next to the first.

Figure 28. Second image beside first image

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", -11, -2);
printer.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

SetPrintArea: If an empty string buffer is sent, then a get only is emulated for compatibility with the directIO related command. Anything else will attempt a set/get for the currently selected PageModeStation. 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 in the following calculations.

```
if x < 0 then x = 0
if y < 0 then y = 0
if x > MaxHorizontalPosition - MinHorizontalPosition then
    x = MaxHorizontalPosition - MinHorizontalPosition
if y > MaxVerticalPosition - MinVerticalPosition
    then y = MaxVerticalPosition - MinVerticalPosition
if width < MinHorizontalPosition then width = MinHorizontalPosition
if height < MinVerticalPosition then height = MinVerticalPosition
if (width + x) > MaxHorizontalPosition then width = MaxHorizontalPosition - x
if (height + y) > MaxVerticalPosition then height = MaxVerticalPosition - y
```

The values for MinVerticalPosition, MaxVerticalPosition, MinHorizontalPosition, and MaxHorizontalPosition are as follows:

- MinVerticalPosition: 6
- MaxVerticalPosition: 800 (1250 for TI8/9)
- MinHorizontalPosition: 15
- MaxHorizontalPosition: 576

Examples:

- ″0,-1,570,600″ is set to: ″0,0, 570,600″
- ″-1,0,800,5000″ is set to: ″0,0,576,800″
- ″-1,6,200,800″ is set to: ″0,6,200,794″
- ″″ is set to: <the current value>

Note: For Models TI8 and TI9, replace ″800″ with ″1250″.

The Page Mode for IBM 4610 Printer can also be used through directIO commands, but this directIO has been deprecated. While in Page Mode, commands are buffered until Page Mode is exited. The buffered commands are sent to the printer after Page Mode completion. The directIO method calls are described in the following section.

Direct IO Methods:

Note: Page mode works through directIO and standard mode. If you use directIO and standard mode you get directIO behavior and not standard.

Syntax: directIO (int cmd, int [ ] data, Object object) throws JposException

DirectIO.CLEARPRINTAREA: Clears the print area defined by directIO.PRINTAREA property.

cmd	com.ibm.jpos.services.DirectIO.CLEARPRINTAREA
data	NULL
object	NULL

DirectIO.PAGEMODEPRINT: Performs various functions as specified by the control value.

Table 59. DirectIO.PAGEMODEPRINT Functions

cmd	com.ibm.jpos.services.DirectIO.PAGEMODEPRINT
data	control
object	NULL

The value of the control as follows:

Table 60. DirectIO.PAGEMODEPRINT Values

Value	Meaning
DirectIO.PTR_PM_PAGEMODE	Enter the page mode.
DirectIO.PTR_PM_PRINTSAVE	Print the print area and save the canvas. Page Mode is not exited. Used for printing repeated pages.
DirectIO.PTR_PM_NORMAL	Print the print area and destroy the canvas and exit page mode.
DirectIO.PTR_PM_CANCEL	Clear the page and exit the page mode without any printing on print area.

Direct IO Properties: The properties of the PageMode support for IBM printer are also accessed using directIO commands. These commands are gotten indirectly. When the properties command is called, the data parameter is set to the supported value that is the printer.

Syntax: directIO (int cmd, int [ ] data, Object object) throws JposException

DirectIO.HORIZONTALPOSITION (read/write): Holds the horizontal start position offset within the print area for the selected directIO.PAGEMODESTATION.

cmd	com.ibm.jpos.services.DirectIO.HORIZONTALPOSITION
data	position
object	NULL

The position[0] has the following meaning.

Value	Meaning
Other value	used to set the horizontal position
X'FF'	returns horizontal position

DirectIO.VERTICALPOSITION (read/write): Holds the vertical start position offset within the print area for the selected directIO.PAGEMODESTATION.

cmd	com.ibm.jpos.services.DirectIO.VERTICALPOSITION
data	position
object	NULL

The position[0] has the following meaning.

Value	Meaning
Other value	used to set the vertical position
X'FF'	returns vertical position

DirectIO.PAGEAREA (read): Holds the page area expressed in the unit of measure given by MapMode. This page area can be different than the print area, and is determined by the hardware capability of the printer. The string consists of ASCII numeric set numbers, separated by commas in the following order: horizontal offset(dx), vertical offset(dy).

For example, if the string is `01,01,20,20`, then the station print area is a rectangular area specified by the top left point at (01,01), and horizontal offset (20), and vertical offset (20).

cmd com.ibm.jpos.services.DirectIO.PAGEAREA

data NULL

object StringBuffer

DirectIO.PAGEMODESTATION (read/write): Sets or returns the print station for subsequent page mode properties.

cmd com.ibm.jpos.services.DirectIO.PAGEMODESTATION

data station

object NULL

The value in station[0] is used to set and return the station in station[0]. The value of station[0] has the following meaning.

Value	Meaning
Other values	used to set the vertical position
X'FF'	only returns current station in station[0]

DirectIO.PRINTAREA: Holds the print area of the last selected print station, expressed in the unit of measure given by MapMode. The maximum print area is the page area. The string consists of ASCII numeric set numbers, separated by commas in the following order: top left coordinate (x1, y1), horizontal offset, vertical offset (dx, dy).

For example, if the string is `01,01,20,20`, then the station print area is a rectangular area specified by the top left point at (01,01), and horizontal offset (20), and vertical offset (20).

cmd com.ibm.jpos.services.DirectIO.PRINTAREA

data NULL

object StringBuffer

DirectIO.PRINTDIRECTION (read/write): Holds the print direction specified by the direction online in the data array.

cmd	com.ibm.jpos.services.DirectIO.PRINTDIRECTION
data	direction
object	NULL

Table 61 explains the meaning of each value.

Table 61. Print directions

Value	Meaning
DirectIO.PTR_PD_LEFTTORIGHT	Print left to right, starting at the top left position of the print area.
DirectIO.PTR_PD_BOTTOMTOTOP	Print bottom to top, starting at the bottom left position of the print area.
DirectIO.PTR_PD_RIGHTTOLEFT	Print right to left, starting at bottom right position of the print area.
DirectIO.PTR_PD_TOPTOBOTTOM	Print top to bottom, starting at the top right position of the print area.

Function of the ErrorEvent with response of ER_CLEAR in the POSPrinter: 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. printNormal2
4. printNormal3
5. ErrorEvent on printNormal3
6. printNormal4
7. printNormal5
8. Respond ErrorEvent with ER_CLEAR
9. printNormal6

All the printNormal events up to, and including printNormal5, are cleared in step 8.

Concurrent MICR/full-image scan

The 4610 Model TI9 will perform a concurrent MICR/full-image scan, however when using a Model TI8 you must do separate passes.

SureOne Version property

Example:

```
<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. Because there is no default value you should 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 62.

Possible Value	Character set
"Standard"	character set 998 (equal to character set 437)
"ChineseTrad"	character set 950
"ChineseSimp"	character set 936
"Korean"	character set 1361
"Japanese"	character set 932

Scale

This section describes the scale device-specific notes for the RS-485 and USB devices. Scale configuration is done through the jpos.xml file.

Configuration options depend on the hardware capabilities. Table 63 details the configuration options supported by the scale models.

Table 63. Scale configuration options

Configuration option	4696	4698	USB OEM
operationMode	✓	✓	✓
displayRequired	✓	✓	✓
centerOfZero	✓	✓	✓
weighMode	✓	✓	✓
enforceZeroReturn	✓	✓	✓
vibrationSensitivity	✓	✓	✓
fiveDigitWeight			✓

Operation mode

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. This property can have the following values:

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).

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.operationMode" type="String" value="US"/>
```

Default Value: US

Supported by the following scale models : IBM 4696 Scanner Scale Model 1, IBM 4698 Scanner Model 2, USB OEM

DisplayRequired

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 scale display will be completely blank. If the value of this resource is *true* but there is no remote display attached to the scale device, the scale will beep six times in rapid succession and will then go offline. This property can have the following values:

true A remote scale display is required.

false A remote scale display is not required.

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.displayRequired" type="Boolean"
      value="false"/>
```

Default Value : false

Supported by the following scale models : 4696, 4698, USB OEM

CenterOfZero

If *true*, indicate center-of-zero with LED.

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.centerOfZero" type="Boolean"
      value="false"/>
```

Default Value : false

Supported by the following scale models : 4696, 4698, USB OEM

WeighMode

This property determines whether the weight is returned in pounds (Avoirdupois or English system) or in kilograms (Metric system). This property can have the following values:

0 Weight is given in pounds. The fiveDigitWeight property specifies the number of digits to return for an English weight (pounds). Four or five digits which represent the weight of the item in hundredths or thousandths of pounds, respectively.

1 Weight is given in kilograms. All scales return five digits which represent the weight of the item in thousandths of kilograms.

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.weighMode" type="String" value="0"/>
```

Default Value : 0

Supported by the following scale models : 4696, 4698, USB OEM

EnforceZeroReturn

This property specifies whether zero protection is required. With zero protection enabled, the scale will not answer weight requests if:

- A negative weight value is indicated on the display prior to placing the item for weighing on the scale. The scale must be zero-adjusted (reset to zero) before weight requests will be answered.
- An item is left on the scale for four minutes. The item must be removed, allowing the scale to return to zero, before weight requests will be answered.

If either of these conditions exists and the value of this property is *true*, no weight data can be transmitted to the host until a rezeroing occurs.

This resource can have the following values:

true enforceZeroReturn is enabled.

false enforceZeroReturn is disabled.

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.enforceZeroReturn"
type="Boolean"  value="false"/>
```

Default Value : false

Supported by the following scale models : 4696, 4698, USB OEM

VibrationSensitivity

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.

This resource can have the following values:

0 Normal

1 Low

2 Very Low

3 Ultra low

Example:

```
<prop name="com.ibm.jpos.sdi.config.Scale.vibrationSensitivity" type="String"
     value="0"/>
```

Default Value: 0

Supported by the following scale models : 4696, 4698, USB OEM

FiveDigitWeight

This property specifies the number of digits to return for an English weight (pounds).

true Five digits of weight data are returned, implying units of thousandths of pounds.

false Four digits of weight data are returned, implying units of hundredths of pounds.

Example:

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

Default Value : *true*

Supported by the following scale models : USB OEM

USB OEM Scale support

The IBM JavaPOS supports new USB Scale devices that comply with USB OEM Point-of-Sale Device specifications. The support for these devices requires modification to configuration files. These configuration files currently include entries for known USB OEM Scale devices. For a new device, the configuration files must be modified to include VendorID and ProductID for 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 Scale device.

1. Obtain VendorID and ProductID for your device using a USB utility, such as UsbView.
2. Edit the javaxusb.inf file, located in C:\POS\IBMJPOS\Lib\.
3. In the OEM Scale 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 Scale section for VID = 0x1234, PID = 0x5678
″Add a description for the scale″=
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 scale.

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 Scale 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:
last entry in OEM Scale section:
com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevices.72 = 0x04b3,0x4900
new entry in the OEM Scale section for Vendor ID = 0x1234, ProductID = 0x5678:
com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevices.**73** = **0x1234,0x5678**

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” on page 107.

Scanner

This section describes the scanner device-specific notes for the EIA-232, RS-485 and USB devices. Scanner configuration is done through the jpos.xml file. The following example shows a typical configuration property:

```
<prop name="setEnableCODE39" type="Boolean" value="true"/>
```

This property enables the capability to read Code39 bar codes.

Notes:

1. Each scanner's hardware capabilities determine which symbologies that scanner 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.

Table 64 lists the supported symbologies for IBM scanners.

Table 64. 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			✓	✓		✓		✓	✓	✓
CODE 39	✓	✓	✓	✓		✓	✓	✓	✓	✓
CODE 93			✓	✓		✓		✓	✓	✓
CODE 128			✓	✓		✓	✓	✓	✓	✓
Standard 2 of 5									✓	✓
UCC/EAN128								✓	✓	✓
RSS-14									✓	✓
RSS-Expanded									✓	✓
PDF-417										✓

Table 65 on page 111 lists the configurable parameters for scanners and indicates which parameters are supported by each scanner model.

Table 65. Scanner configurable parameters

Parameter	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
enableLaserOnOffSwitch								✓		
enableProgrammingViaBarcodes							✓		✓	✓
enableSwitchControlledVolumeAdjust					✓	✓	✓	✓		
setBarCodes1	✓	✓	✓	✓	✓	✓	✓	✓		
setBarCodes2	✓			✓		✓	✓	✓		
setBarCodes3	✓						✓	✓		
setBarCodes4	✓						✓	✓		
setBeeperDuration					✓	✓	✓	✓	✓	
setBeeperFrequency					✓	✓		✓	✓	✓
setBeeperVolume					✓	✓	✓	✓		
setCheckModulo		✓	✓	✓						
setCode128ScansPerRead							✓			
setCode39ScansPerRead							✓			
setDecodeAlgorithm					✓	✓	✓			
setDoubleReadTimeOut					✓	✓	✓	✓		
setDTouchMode		✓	✓	✓						
setEAN13ScansPerRead					✓	✓	✓			
setEAN8ScansPerRead					✓	✓	✓			
setEnableCodabar			✓	✓		✓	✓	✓	✓	✓
setEnableCode128			✓	✓		✓	✓	✓	✓	✓
setEnableCode128Supplementals								✓		
setEnableCODE39	✓	✓	✓	✓		✓	✓	✓	✓	✓
setEnableCode39CheckDigit								✓		
setEnableCode93			✓	✓		✓	✓	✓	✓	✓
setEnableEAN_JAN_TwoLabelDecoding						✓	✓	✓		
setEnableGoodReadBeep	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
setEnableInterleaved2of5	✓	✓	✓	✓		✓	✓	✓	✓	✓
setEnableITFCheckDigit								✓		
setEnablePDF417										✓
setEnableRSS14									✓	✓
setEnableRSS_Expanded									✓	✓
setEnableStandard2of5									✓	✓
setEnableUCC_EAN128								✓	✓	✓
setEnableUPCAE_EANJAN813	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
setEnableUPCD1D5			✓		✓	✓	✓	✓		
setEnableUPC_A_CheckDigit								✓	✓	✓
setEnableUPC_A_To_EAN13Expansion					✓	✓	✓	✓	✓	
setEnableUPC_E_CheckDigit								✓	✓	✓

Table 65. Scanner configurable parameters (continued)

Parameter	1520	4500	4501	4685	4696	4697	4698	USB OEM	AnyPlace Kiosk Line	AnyPlace Kiosk Omni
setEnableUPC_E_To_EAN13Expansion					✓	✓	✓	✓		
setEnableUPC_E_To_UPC_AExpansion							✓	✓		✓
setEnableVerificationUPC_A_EAN13_ fiveDigit					✓	✓	✓	✓		
setEnableVerificationUPC_A_EAN13_ fourDigit					✓	✓	✓	✓		
setEnable_2_DigitSupplementals				✓			✓	✓	✓	
setEnable_5_DigitSupplementals				✓			✓	✓	✓	
setITFLength1	✓					✓	✓	✓	✓	✓
setITFLength2						✓	✓	✓	✓	✓
setITFLengths								✓	✓	✓
setITFLengthSpecifiedTwo						✓	✓	✓		
setITFScansPerRead							✓			
setLaserTimeOut					✓	✓	✓	✓		
setLED_GoodRead_Duration								✓		
setMotorTimeOut					✓	✓	✓	✓		
setSecurityLevelForInStore								✓		
setSTFLength1									✓	✓
setSTFLength2									✓	✓
setSTFLengths									✓	✓
setStoreScansPerRead					✓	✓	✓			
setSupplementals								✓		✓
setSupplementalsSecurityLevel									✓	✓
setUPCAScansPerRead					✓	✓	✓			
setUPCDScansPerRead					✓	✓	✓			
setUPCEScansPerRead					✓	✓	✓			

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

Property Type: boolean

Default: false

Models Supported: 4696, 4697, 4698, AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

This property controls whether or not the scanner can be programmed using the manufacturer-supplied programming bar codes.

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 four properties which determine what combination of bar code types a scanner will recognize. When configured in a specific mode, a scanner recognizes and returns all bar code types that are associated with that mode. Table 66 lists the values that can be given for the setBarCodes1 property for both RS-485 and USB-attached scanners.

Table 66. setBarCodes1 values

Resource Value	Bar Code Types
0 - GROUP_NONE	None
1 - GROUP_UPC_EAN_ITF	UPCAE_EANJAN813 Interleaved2of5
2 - GROUP_UPCAED	UPCAE_EANJAN813 UPCD1D5
3 - GROUP_UPC_EAN_CODE128	UPCAE_EANJAN813 Code128
4 - GROUP_UPC_EAN_CODE93	UPCAE_EANJAN813 Code93
5 - GROUP_UPC_EAN_CODE39	UPCAE_EANJAN813 CODE39
6 - GROUP_UPC_EAN_CODABAR	UPCAE_EANJAN813 Codabar
7 - GROUP_UPC_EAN_2_5_CODABAR	UPCAE_EANJAN813 Codabar Enable_2_DigitSupplementals Enable_5_DigitSupplementals

setBarCodes2

Property Type: byte

Default: 0

Scanner models supported: 1520, 4685, 4697, 4698, USB OEM

This property is one of four propertiess which determine what combination of bar code types a scanner will recognize. 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 four properties which determine what combination of bar code types a scanner will recognize. 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 four properties which determine what combination of bar code types a scanner will recognize. 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 EAN13 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 EAN8 labels.

This resource can have the following values:

1 One scan

2 Two scans

3 Three scans

4 Four scans

setEnableCodabar
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 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, 4698, USB OEM

Enable or disable the Code39 barcodes.

setEnableEAN_JAN_TwoLabelDecoding

Property Type: boolean

Default: false

Models Supported: 4697, 4698, USB OEM

Enable or disable the EAN/JAN Two-label decoding.

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

When these barcodes are enabled the scan mode is changed to allow 2D barcodes reading. For this reason, omnidirectional reading is not available in this mode.

setEnableRSS14

Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the RSS14 barcodes.

setEnableRSS_Expanded
Property Type: boolean

Default: false

Models Supported: AnyPlace Kiosk Line Scanner, AnyPlace Kiosk Omni Scanner

Enables or disables the RSS 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.

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.

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.

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.

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, UPE, EAN/JAN8, and EAN/JAN13. When enabled, supplementals are optional for all these symbologies.

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 will read. 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 will be 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: 30

Models Supported: AnyPlace Kiosk Line scanner, AnyPlace Kiosk Omni scanner

This property specify the security level for UPCA, UPCE, EAN/JAN8, AND EAN/JAN13 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.

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

USB OEM Scanner support

The IBM JavaPOS supports new USB scanner and scale devices that comply with USB OEM Point-of-Sale Device 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 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:
last entry in OEM Scanner section:
com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevices.72 = 0x04b3,0x4900
new entry in the OEM Scanner section for Vendor ID = 0x1234, ProductID = 0x5678:
com.ibm.posj.bus.hid.javaxusb.factory.knownPosDevices.**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 will 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 will not work correctly if these are enabled.

ToneIndicator

When you are using the tone device on an EIA-attached 4610 TM6/7 or TF6/7, you must configure a jpos entry for both the Printer and the ToneIndicator. Otherwise the ToneIndicator will not work.

OPOS

This section provides device-specific notes for OPOS.

CheckScanner

Three DirectIO commands are supported for the IBM 4610 CheckScanner device:

- Set Check side (TI8/TI9)
- Get Check Side (TI8/TI9)
- Get Image Scan Quality (TI9 only)

The DirectIO syntax is:

```
LONG DirectIO (LONG Command, LONG * Data, BSTR * String)
```

Table 67. 4610 CheckScanner supported commands for DirectIO

Command	LONG *	BSTR *	Comments
CHK_DIO_SET_CHECK_SIDE_CMD	IN Data	N/A	Sets the side of the check to scan; that is flips it. Assuming a check is initially inserted upside down ready for scanning then this is considered to be logical side 1 at the first insertion. After each flip the new current side is tracked by the service. Set the value in Data to CHK_DIO_SIDE_2 (2) to flip the check to logical side 2. Set the value in Data to CHK_DIO_SIDE_1 (1) to flip the check to logical side 1. Set the value in Data to CHK_DIO_OPPOSITE (3) to flip the check to the opposite logical side.
CHK_DIO_GET_CHECK_SIDE_CMD	OUT Data	N/A	Returns the current logical check side as tracked by the service. The value is returned in the Data parameter. Values Returned: CHK_DIO_SIDE_1 (1) CHK_DIO_SIDE_2 (2)
CHK_DIO_SCANNER_IMAGE_QUALITY_CMD	OUT Data	N/A	Returns the image scan quality of the ImageData property contents associated with the latest DataEvent returned to the application. The value is returned in the Data parameter. Values Returned: CHK_IMAGE_QUALITY_OK (0) CHK_IMAGE_QUALITY_BAD (1)

All three methods require that the device is enabled.

Set Check Side

- If the side the check is on and the side requested are the same, the method does nothing and returns OPOS_SUCCESS.
- A check must be inserted fully in the DI station, or the method will fail.

Get Check Side

The method returns the current logical check side tracked whether or not a check is inserted.

Get Image Scan Quality

- The method fails if the printer model is not a TI9 with microcode version of at least 0x68.
- The quality referred to is the optical quality of the image when it was scanned.

Keyboards

This section provides device-specific notes for keyboards.

CANPOS keyboard

See “CANPOS keyboard” on page 85.

USB system-attached keyboard

The correct device driver is automatically installed for the USB System Attached Keyboard on Windows XP. If you select **IBM Alphanumeric Point of Sale Keyboard** and indicate that it is attached to a USB port, the system reboots. 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 booted *again* before the new driver can be used.

SureMark printer

TI8 and TI9 firmware download

When a system is first initialized, this package checks the firmware level of the device and upgrades the firmware if necessary. With the TI8/9, the firmware file is large and the initial download could take several minutes.

Check scanning support

DirectIO commands are available to support check scanning on the SureMark TI8/9 POS Printers. Table 68 lists the check scanning supported commands for DirectIO.

Table 68. Check scanning supported commands for DirectIO

Command	LONG *	BSTR *	Comments
CS_MICR_READ_WITH_SCAN (201)	Data	N/A	This value is used to determine if a MICR read will occur in the same pass as the check scanner. Set value in 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 set the Data to the current value. 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.
CS_COMPRESSION_FORMAT (202)	data	N/A	This value is used to set the compression format when stored or retrieved. Format of an image once stored does not change. To request the current value set Data to GET_VALUE(-1) , which will set Data to the current value. 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 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)
CS_SCAN_DOCUMENT (203)	N/A	String	This command is 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 will contain the width and length of the scanned document separated by a comma if successful.

Table 68. Check scanning supported commands for DirectIO (continued)

Command	LONG *	BSTR *	Comments
CS_STORE_DOCUMENT (204)	Data	String	This command is to store all or part of the last document scanned. Document will be stored in the format specified by compression format. The value in Data will set the area or areas to store. After the command returns, Data will contain the first index of the storage area in the printer. 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. For Data: **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” for additional information.
CS_RETRIEVE_BY_INDEX (205)	Data	String	This command retrieves the image stored at the specified location 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.
CS_RETRIEVE_BY_TAGNAME (206)	N/A	String	This command retrieves the image stored with the tagData specified in String and stores it in String. The String data format is set based on BinaryConversion. 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.
CS_ERASE_STORAGE (207)	N/A	N/A	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 storage 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 terminated by null character, maximum 100 characters

String will be interpreted based upon BinaryConversion. All offsets are based upon MapMode. If MapMode is set to _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.

When storage method = CS_STORE_ENTIRE(0), all data defined within x0, y0, dx0, dy0 is stored. The command parameters, x1, x2, x3, y1, y2, y3, dx1, dx2, dx3, and dy1, dy2, dy3 should not be included.

When storage method = 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 the name/address and the MICR blocks are saved. The command parameters, x0, x1, x2, x3, y0, y1, y2, y3, dx0, dx1, dx2, dx3, and dy0, dy1, dy2, dy3 should not be included.

When storage method = 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. The command parameters, x0, x1, x2, x3, y0, y1, y2, y3, dx0, dx1, dx2, dx3, and dy0, dy1, dy2, dy3 should not be included.

When storage method = CS_STORE_UD(3), this method allows the user to define sub-blocks of data to store. All command parameters 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 29 on page 137). 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.

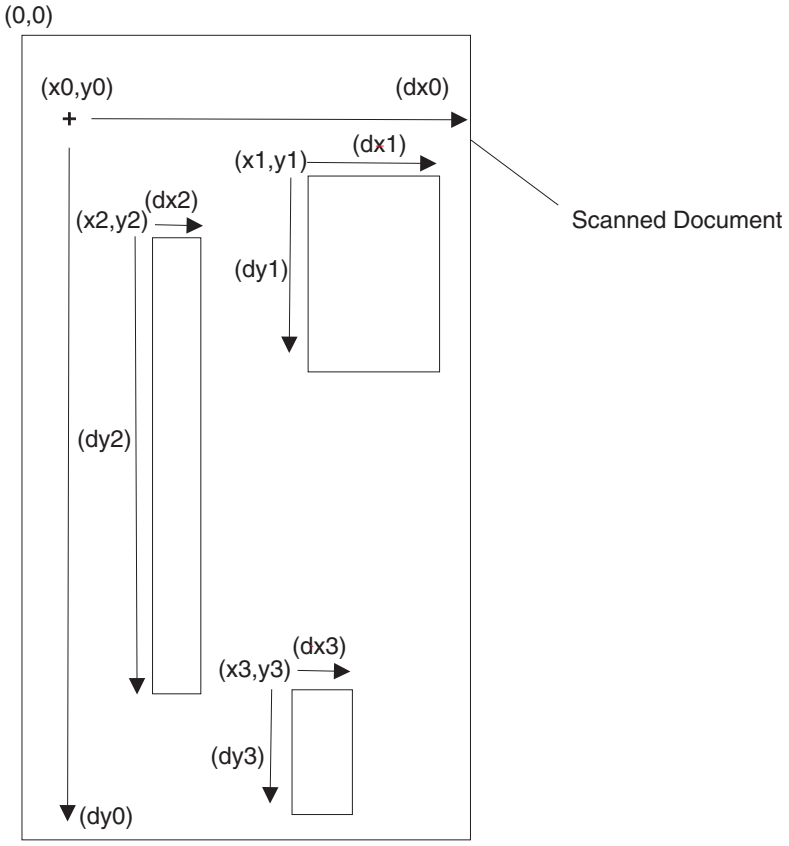


Figure 29. Scanned image organization

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 in the x direction (in hundredths of inch increments), 2 bytes
  - Height of image in the y direction (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.

Table 69. Page Mode supported commands for DirectIO

Command	LONG *	BSTR *	Comments
PM_CLEAR_PRINT_AREA (101)	N/A	N/A	Clears the print area defined in the current PrintArea
PM_PAGE_MODE_PRINT (102)	Data	N/A	Control 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.
PM_HORIZONTAL_POSITION (103) PM_VERTICAL_POSITION (104)	N/A	N/A	Sets and returns the position for the horizontal or the vertical print for the currently selected PageModeStation. If the value sent in position 0 isGETVALUE(-1) the current position is returned. Anything other than GETVALUE will attempt to set/get the position and the value that will be used 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.
PM_PAGE_AREA (105)	N/A	String	Returns the Page area of the printer inside the StringBuffer object for the currently selected PageModeStation. The format is “x,y”, where x is the horizontal width and y is the vertical width.
PM_PAGE_MODE_STATION (106)	Data	N/A	Sets and returns the current station via position 0 or the data int array. If the value sent in position 0 is GETVALUE(-1) the current station is returned. Anything other than GETVALUE will attempt to set/get the station and the value that will be used is returned in position 0.
PM_PRINT_AREA (107)	N/A	String	Sets and returns the desired Print Area. If an empty StringBuffer is sent then a get only is emulated. Anything else will attempt a set/get 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, and dy is the vertical offset.

Table 69. Page Mode supported commands for DirectIO (continued)

Command	LONG *	BSTR *	Comments
PM_PRINT_DIRECTION (108)	value	N/A	Valid values for the currently selected PageModeStation are: GETVALUE (-1) - get only PM_PD_LEFTTORIGHT (1) PM_PD_BOTTOMTOTOP (2) PM_PD_RIGHTTOLEFT (3) PM_PM_TOPTOBOTTOM (4)

DirectIO commands also are 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.

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 code pages to the SureMark printers that support DBCS is explained in Part 3, “Keyboards and code pages,” on page 237.

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 and 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 to some degree, provided we make clear a couple assumptions. First, 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 will feed when it has reached the end of the line.

Second, text alignment is based on the fact that a fixed number of characters will 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 will use a width of half of the height and then adjust it for a best fit within the valid range.

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 with cause the enable to fail on the second application. If using the tone indicator from two applications, it is suggested that the applications 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 70 on page 141 lists the characters for each code set.

Table 70. 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 will be 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 certain value, 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 71 lists the character pairs for encoding the special characters:

Table 71. 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
{	{{

CashDrawer support

With the addition of cash drawer support for USB-attached SureMark printer, it is clear that some OEM cash drawers are wired opposite of IBM cash drawer for determining the cash drawer status, thus the Signals Reversed check box in the configuration utility. The USB attached SureMark printer has standardized on the IBM cash drawer wiring. However, it was determined that the EIA-232 attached SureMark printers was coded using the opposite OEM wiring. The driver has been changed to match the USB attached SureMark driver starting with release 1.7.1. Therefore, it may be necessary to reconfigure your EIA-232 attached SureMark printer cash drawer and set or clear the Signals Reversed check box.

Flash memory

All OPOS access to the Flash memory in the SureMark printer is by using DirectIO() methods. To support the flash memory in the 4610 printer, the following DirectIO Commands are added to the SureMark OPOS Service Object for the RS-485, USB and EIA-232 interfaces. The following DirectIO() commands are provided:

WRITE_FLASH_MEMORY (0x10): 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* is a 32 bit

number, ***in little-endian format***, indicating the record number and *n1* and *n2* is a 16 bit number, ***in little-endian format***, indicating the number of data bytes to be written.

When *AsyncMode* is set to *true*, the data will be queued to the printer. Any error associated with this write will cause a DirectIoEvent with the EventNumber set to **DIRECTIO_FLASH_ERROR (0xFE)**. If the record number is beyond the maximum records supported, **pData** will be 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** will be set to **DIRECTIO_FLASH_ERROR_REASON_TOO_LONG (0x2D).** When *AsyncMode* is *false*, this command will wait until the memory is written to the printer (or an error condition) before returning control to the application. If there is an error, ResultCodeExtended will be set with either of the two values, or with the POSSWIN internal error.

READ_FLASH_MEMORY (0x11): Read the flash memory record number. The format of the data to be read is **'r1.r2.r3.r4.**'. Where *r1*, *r2*, *r3*, and *r4* is a 32 bit number, ***in little-endian format***, indicating the record number.

When *AsyncMode* is set to *true*, the data will be returned to the application by a DirectIoEvent with the EventNumber set to **DIRECTIO_FLASH_DATA (0xFD)**. Any error associated with this read will cause a DirectIoEvent with the EventNumber set to **DIRECTIO_FLASH_ERROR (0xFE)**. If the record number is beyond the maximum records supported, **pData** will be set to **DIRECTIO_FLASH_ERROR_REASON_OUT_OF_RANGE (0x2F).** If there is not response to the request within 5 seconds, **pData** will be 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** will be set to **DIRECTIO_FLASH_ERROR_REASON_TOO_LONG (0x2D).** When *AsyncMode* is *false*, this command will wait until the memory is read from the printer (or an error condition) before returning the data to the application. The record will be returned in the pString field of the DirectIO call. If there is an error, ResultCodeExtended will be set with any one of these three values, or with the POSSWIN internal error.

QUERY_FLASH_SIZE (0x12): Returns the size of memory. The value will be returned in the pData field of the DirectIO call.

QUERY_MAXIMUM_RECORDS (0x13): Returns the maximum number of records. This number will be calculated by dividing the maximum printer memory by the application requested memory size. The value will be returned in the pData field of the DirectIO call.

SET_RECORD_LENGTH (0x14): 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 will not automatically erase the memory. The value should be set in the pData field of the DirectIO call.

ERASE_FLASH_MEMORY (0x15): Erases all data stored in the flash memory on the 4610 printer.

GET_RECORD_LENGTH (0x16): Retrieves the number of bytes for each record written to the flash memory. The value will be zero if it has not yet been set after an erase.

SureOne

Keyboard

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

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 will be 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*.

DBCS printer limitations

For a DBCS printer, the printer code page is assumed to be the same as the locale of Windows.

HardTotals

The SureOne A04 and A05 does not support Hard Totals. No non-volatile RAM is available on these models.

OPOS Gateway

The OPOS Gateway provides all the function provided by JavaPOS. However, 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 will convert the XML string to a JavaPOS object during the DirectIO call. You should use the provided XML Converter utility to create and parse the XML data.

Supported DirectIO calls

Table 72 and Table 73 lists the DirectIO calls that are supported in OPOS Gateway.

Table 72. Printer DirectIO calls supported in OPOS Gateway

Printer Commands	Data[]/ LONG *	Object/ BSTR *	Comments
DOWNLOAD_PROP_FONT_ID - use Objects	N/A	DirectIOFontInfo	Download Proportional Font, the object holds data describing the font.
DOWNLOAD_NON_PROP_FONT_ID - use Objects	N/A	DirectIOFontInfo	Download Non-proportional Font, the object holds data describing the font.
GET_FONT_INFO_ID	N/A	PrintGetDirectIOFontInfo	Get Font Information
DOWNLOAD_DBCS_FONT_ID	N/A	DBCSFontInfo	Download DBCS Font to the printer, the object contains the information of the font.

Table 73. Fiscal Printer DirectIO calls supported in OPOS Gateway

Fiscal Printer Commands	Data[]/ LONG *	Object/ BSTR *	Comments
FISCAL_WRITE - uses a byte array	FISCAL_WRITE	byte[]	Write to Fiscal Printer
FISCAL_READ - uses a byte array	FISCAL_READ	byte[]	Read from Fiscal Printer
FISCAL_INFORMATION - uses a Object	FISCAL_INFORMATION	Object FiscalInformation	Retrieve Fiscal information

Provided DirectIO XML Converter access and calls

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

The following list describes all of the methods provided by the DirectIOXMLConverter COM interface:

- Function GetByteArray(XML As String, byteArray() As Byte) As String
  - converts an XML string containing an encoded byte array into an array of bytes
  - returns “Success” when it works or a string with the reason if it fails
  - used to do Fiscal_Read and Fiscal_Write DirectIO calls
- Function GetByteArrayXML(byteArray() As Byte) As String
  - converts an array of bytes into an XML string that contains the array of bytes encoded
  - used to do Fiscal_Read and Fiscal_Write DirectIO calls
  - The byte array passed must have the correct number of elements for a Fiscal_Read. 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 will be sent to the Fiscal Printer so the length is based on the Fiscal command.)
- Function GetDBCSFontInfo(XML As String, FileName As String) As String
  - extracts the FileName field from a DBCSFontInfo object encoded in an XML string
  - returns “Success” when it works 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 “Success” when it works 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
  - returns an XML string containing an encoded DBCSFontInfo object or empty string if it fails
  - used to download a DBCS font to the POS Printer
- 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 “Success” when it works 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 “Success” when it works 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
  - returns an XML string containing an encoded DirectIOFontInfo object or empty string if it fails
  - used to download a Proportional or Non-Proportional font to the POS Printer
- Function GetFiscalInformation(XML As String, FiscalCountry As Long, FiscalPowerInterrupted As Boolean, FiscalVersion As Long) As String
  - extracts the FiscalCountry, FiscalPowerInterrupted and FiscalVersion fields from a FiscalInformation object encoded in an XML string
  - returns “Success” when it works or a string with the reason if it fails
  - used to get the FiscalCountry, FiscalPowerInterrupted, and FiscalVersion data from the Fiscal Printer (GetFiscalInformationObject can also be used)
- Function GetFiscalInformationObject(XML As String, obj As FiscalInformation) As String
  - converts an XML string containing an encoded FiscalInformation object into a FiscalInformation object
  - returns “Success” when it works or a string with the reason if it fails
  - used to get the FiscalCountry, FiscalPowerInterrupted, and FiscalVersion data from the Fiscal Printer using the FiscalInformation object (GetFiscalInformation can also be used)
- 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 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 o
  - 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
  - used to get the Font Info data from the Fiscal Printer (GetPrintGetDirectIOFontInfoObject can also be used)
- Function GetPrintGetDirectIOFontInfoObject(XML As String, obj As PrintGetDirectIOFontInfo) As String
  - 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
  - used to get the Font Info data from the Fiscal Printer (GetPrintGetDirectIOFontInfo can also be used)
- 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

The public classes and the public data members of each of the COM objects in DirectIOXMLConverter are

- DBCSFontInfo
  - filename as String
- DirectIOFontInfo
  - Station as Long
  - FontNumber as Long
  - FileName as String
  - Len as Long
- FiscalInformation
  - FiscalCountry as Long
  - FiscalPowerInterrupted as Boolean
  - FiscalVersion as Long
- PrintGetDirectIOFontInfo
  - receiptCharacterSet(0 to 3) as Long

    Note: array of Longs
  - 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

4610 Tx6/7 POSPrinter attached ToneIndicator (EIA-232 bus only)

When you are using a 4610 TM6/7 or TF6/7 printer attached ToneIndicator with EIA-232 bus, you must configure a jpos entry for both the printer and the ToneIndicator, otherwise, the ToneIndicator won't populate.

Part 2. Programming reference

Chapter 5. Supported properties and methods

This chapter lists properties, methods, and events for both the OPOS and JavaPOS interfaces. The standard UnifiedPOS property and method names are used to reduce redundancy and improve readability.

The following properties are used only in OPOS:

- BinaryConversion
- OpenResult
- ResultCode
- ResultCodeExtended

Some properties have different names in OPOS. Table 74 lists the UnifiedPOS and OPOS names for these properties.

Table 74. 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

In this chapter, each device class has multiple tables. The first table lists all of the supported hardware. In some cases, each type of hardware has a numeric value associated with it. The subsequent tables list the properties, methods and events supported. In the **JavaPOS** and **OPOS** columns, a specific hardware device may be referenced using the numeric value from the first table.

CashDrawer

Table 75. CashDrawer supported devices

Device	Connectivity
1. SurePOS 500/600 family built-in cash drawer	EIA-232
2. SurePOS 700 family built-in cash drawer	USB, 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

Table 76. CashDrawer common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	
CapCompareFirmwareVersion	False	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 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	Not supported
Claimed	All	All
DataCount	Not supported	Not supported
DataEventEnabled	Not supported	Not supported
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
PowerNotify	All	All
PowerState	All	All
State	All	All

Table 77. CashDrawer specific properties

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

Table 78. CashDrawer common methods

Method	JavaPOS and OPOS Gateway	OPOS
compareFirmwareVersion	Not supported	Not supported
checkHealth	All	All except SurePOS 500/600 family built-in cash drawer
claim	All	All
clearInput	Not supported	Not supported
clearOutput	All	Not supported
close	All	All
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 79. CashDrawer specific methods

Method	JavaPOS and OPOS Gateway	OPOS
openDrawer	All	All
waitForDrawerClose	All	All

Table 80. CashDrawer events

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

CheckScanner

Table 81. CheckScanner supported devices

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

Table 82. CheckScanner common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	All	Supported
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	Supported
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	Supported
Claimed	All	Supported
CompareFirmwareVersion	Not supported	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 83. CheckScanner specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapAutoContrast	False	False
CapContrast	True	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 83. CheckScanner specific properties (continued)

Property	JavaPOS and OPOS Gateway			OPOS	
RemainingImagesEstimate		TI8/TI9	TI9	TI8/TI9	TI/9
(Estimates when memory is empty.)	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 84. CheckScanner 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
clearOutput	All	Supported
close	All	Supported
directIO	All	Supported (see Table 67 on page 132)
open	All	Supported
release	All	Supported
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 85. CheckScanner 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 85. CheckScanner 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	All

Table 86. CheckScanner events

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

FiscalPrinter

Note: Fiscal printers are not supported in the current release of OPOS.

Device	Connectivity
1. IBM SureMark Printer Fiscal Printer Models Kx3	EIA-232
2. IBM SureMark Printer Fiscal Printer Models Kx4	EIA-232
3. IBM SureMark Printer Fiscal Printer Models Kx5	EIA-232
4. IBM SureMark Printer Fiscal Printer Models Kx3	RS-485
5. IBM SureMark Printer Fiscal Printer Models Kx5	RS-485
6. IBM SureMark Printer Fiscal Printer Models Kx5	USB
7. IBM SureMark Printer Fiscal Printer Models Gx3	RS-485
8. IBM SureMark Printer Fiscal Printer Models Gx3	USB
9. IBM SureMark Printer Fiscal Printer Models Gx5	RS-485
10. IBM SureMark Printer Fiscal Printer Models Gx5	USB
11. IBM SureMark Printer Fiscal Printer Models 3xA	RS-485

Table 87. FiscalPrinter common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	Not supported
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	Not supported
Claimed	All	Not supported
DataCount	Not supported	Not supported
DataEventEnabled	Not supported	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 88. FiscalPrinter 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
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

Table 88. FiscalPrinter specific properties (continued)

Property	JavaPOS	OPOS
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 & 2 - false 3 - true	Not supported
CapSlpPresent	1 & 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

Table 88. FiscalPrinter specific properties (continued)

Property	JavaPOS	OPOS
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
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 89. FiscalPrinter common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	Not supported
claim	All	Not supported
clearInput	Not supported	Not supported
clearOutput	Not supported	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 341)	E_ILLEGAL
updateFirmware	E_ILLEGAL	E_ILLEGAL
updateStatistics	E_ILLEGAL	E_ILLEGAL

Table 90. FiscalPrinter specific methods

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

Table 91. FiscalPrinter events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All - yes See “FiscalPrinter” on page 57 for details.	Not supported
ErrorEvent	Not supported	Not supported
OutputCompleteEvent	Not supported	Not supported
StatusUpdateEvent	Not supported	Not supported

HardTotals

Device	Connectivity
1. 4674 NVRAM	Integrated
2. 4694 NVRAM	Integrated
3. SurePOS 700 Series NVRAM	Integrated
4. SureOne built-in NVRAM	Integrated (OPOS only)

Table 92. HardTotals common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_NONE	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	Not supported	Not supported
DeviceEventEnabled	Not supported	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	All	All
OutputID	Not supported	All
PowerNotify	PN_DISABLED	All
PowerState	PS_UNKNOWN	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	All	All
ResultCodeExtended	All	All
State	All	All

Table 93. HardTotals specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapErrorDetection	Not supported	Not supported

Table 93. HardTotals specific properties (continued)

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

Table 94. HardTotals common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	Not supported	Not supported
clearOutput	Not supported	Not supported
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 95. HardTotals specific methods

Method	JavaPOS and OPOS Gateway	OPOS
beginTrans	Not supported	Not supported
claimFile	All	All
commitTrans	Not supported	Not supported
create	All	All
delete	All	All
find	All	All
findByIndex	All	All
releaseFile	All	All
read	All	All
recalculateValidationData	Not supported	Not supported
rename	Not supported	Not supported
rollback	Not supported	Not supported
setAll	All	All
validateData	Not supported	Not supported

Table 95. HardTotals specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
write	All	All

Table 96. HardTotals events

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

Keylock

Table 97. Keylock supported devices

Device		Connectivity
Two-position	1. Alphanumeric POS Keyboard keylock	PS/2, USB, RS-485
	2. Alphanumeric POS Keyboard keylock with mouse pointer	PS/2
	3. 50-key Keyboard keylock	USB, RS-485
	4. 133-key Keyboard keylock	USB, RS-485
	5. 4820 keylock	USB, RS-485
Three-position	SureOne keylock	Integrated
Four-position	1. Alphanumeric POS Korea Keyboard keylock	PS/2, USB, RS-485
	2. Keyboard V keylock	USB, RS-485
	3. 4685-KC1	RS-485 (OPOS only)
	4. 4685-K01	RS-485 (OPOS only)
	5. 4685-K02 (Ultra 7)	USB, RS-485
Five-position	4674 Built-in	RS-485
Six-position	4685–K02 (Ultra7) keyboard keylock	USB, RS-485

Table 98. Keylock common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
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 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	Not supported	Not supported
DataEventEnabled	Not supported	Not supported
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All

Table 98. Keylock common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	Not supported
OutputID	Not supported	Not supported
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
State	All	All

Table 99. Keylock specific properties

Property	JavaPOS and OPOS Gateway	OPOS
KeyPosition	All	All
PositionCount	All	All

Table 100. Keylock common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	Always shareable	Always shareable
clearInput	Not supported	Not supported
clearOutput	Not supported	Not supported
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	Always shareable	Always shareable
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 101. Keylock specific methods

Method	JavaPOS and OPOS Gateway	OPOS
waitForKeylockChange	All	All

Table 102. Keylock events

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

LineDisplay

Table 103. LineDisplay 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	1. Character Graphic (C/G) APA	EIA-232
	2. Character Graphic (C/G) APA	USB, RS-485
	3. PLU keyboard APA	USB, RS-485

Table 104. LineDisplay common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_NONE for: SBCS Device 1 EIA-232 DBCS Device 1 EIA-232 PR_STANDARD for all other LineDisplay	PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	Not supported	Not supported
DataEventEnabled	Not supported	Not supported
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	PN_DISABLED	All

Table 104. LineDisplay common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 105. LineDisplay specific properties

Property	JavaPOS and OPOS Gateway	OPOS
BlinkRate	Not supported	Not supported
CapBlink	Not supported except 10 (set CB_BLINKALL)	Not supported except 8 (set CB_BLINKALL)
CapBitmap	Not supported	Not supported
CapBlinkRate	Set to `DISP_CB_NOBLINK` in level 1.7	Not supported
CapBrightness	Not supported except 1 & 10 (Supported on EIA-232 integrated and external line displays)	Not supported except 1 & 10 (Supported on EIA-232 integrated and external line displays)
CapCharacterSet	All (values based on HW)	All (values based on HW)
CapCursorType	See Table 109 on page 174.	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: set false
CapHMarquee	Not supported	Not supported
CapICharWait	Not supported	Not supported
CapMapCharacterSet	Not supported	Not supported
CapReadBack	Not supported	Not supported
CapReverse	Not supported	Not supported
CapScreenMode	8	Not supported
CapVMarquee	Not supported	Not supported
CharacterSet	All (values based on HW)	All (values based on HW)
CharacterSetList	All (values based on HW)	All (values based on HW)
CurrentWindow	All	All

Table 105. LineDisplay specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
Columns	All	All
CursorColumn	All	All
CursorRow	All	All
CursorType	See Table 109 on page 174. Not supported directly as property, but supported using DirectIO method for EIA-232 external LineDisplay (All points addressable (APA))	Not supported
CursorUpdate	All	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
CustomDSGlyphList	Not supported	Not supported
CustomSSGlyphList	Not supported	Not supported
DeviceBrightness	Supported on EIA-232 integrated and external LineDisplays	0 or 100% only except on 1 & 10 (0 to 100%)
DeviceColumns	All	All

Table 105. LineDisplay specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
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	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	Not supported
MapCharacterSet	Not supported	Not supported
MarqueeFormat	Not supported	Not supported
MarqueeRepeatWait	Not supported	Not supported
MarqueeType	Not supported	Not supported
MarqueeUnitWait	Not supported	Not supported
MaximumX	Not supported	Not supported
MaximumY	Not supported	Not supported
Rows	All	All
ScreenMode	**1, 2, 3, 6, 7** 0 **8** 0, 1, 2 **9** 0	Not supported

Table 105. LineDisplay specific properties (continued)

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

Table 106. LineDisplay common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	Not supported	Not supported
clearOutput	Not supported	Not supported
close	All	All
compareFirmwareVersion	Not supported	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	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 107. LineDisplay specific methods

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

Table 107. LineDisplay specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
displayBitmap	Not supported	Not supported
displayText	All	All
displayTextAt	All	All
readCharacterAtCursor	Not supported	Not supported
refreshWindow	Not supported	All
scrollText	All	All
setBitmap	Not supported	Not supported
setDescriptor	Supported only for USB-VFD LineDisplay	All except 1 and 10

Table 108. LineDisplay events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All	All
StatusUpdateEvent	All	All

Table 109. CapCursor Type and Cursor Type Properties

Cursor type	LineDisplay VFD USB	LineDisplay LCD USB (50 key and external)	LineDisplay VFD 2x20 EIA-232 (integrated and external)	LineDisplay APA EIA-232
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.				

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()

The EIA232 Character Graphic (C/G) Display jpos.xml entry has the following properties:

```
<prop name="com.ibm.jpos.services.sdi.config.LineDisplay.NumberOfRows"
     type="String" value="2"/>
<prop name="com.ibm.posj.bus.rs232.lineDisplayId" type="String" value="APA/>
```

And the EIA232 VFD 2X20 LineDisplay display has the following jpos.xml entry:

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

It's definition is as follows:

LineDisplayID
: Distinguishes between EIA232 APA and EIA232 VFD displays when creating the handleImp for them because although they are on the same data bus, they use different hardware commands. This property is useful only for internal code purposes.

CursorState
: Initializes the EIA232 LineDisplay VFD cursor state to On or Off. This is done only at initialization time.

NumberOfRows
: Initializes the EIA232 LineDisplay APA to a selected number of rows. Options are: 2, 4, or 5.

Character sets supported by LineDisplay devices

Table 110 shows the country character sets supported by the LineDisplay devices. 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 110. Character sets supported by LineDisplay devices

Character set	LineDisplay device						
	APA (EIA-232)	**VFD Integrated & Distributed EIA-232**	**VFD 1-Sided (USB)**	**LCD & 50-Key LineDisplay**	**APA (USB)**	**VFD 1-Sided & 2-Sided (RS-485)**	**APA (RS-485)**
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)					✓		
950 (Traditional Chinese)	✓						
1361 (Korean)	✓						

MICR

Note: See “MICR” on page 73 for additional information.

Table 111. MICR supported devices

Device	Connectivity
1. 4610 printer family MICR	EIA-232
2. 4610 printer family MICR	RS-485
3. 4610 printer family MICR	USB
4. Model 4R printer MICR	RS-485
5. Model 4R printer MICR	USB

Table 112. MICR common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	All	All
DataEventEnabled	All	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	All	PN_DISABLED
PowerState	All	PS_UNKNOWN
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 113. MICR specific properties

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

Table 114. MICR common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	All	All
clearOutput	Not supported	Not supported
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 115. MICR specific methods

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

Table 116. MICR events

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

MSR

Table 117. MSR supported devices

Device		Connectivity
ISO (3-track)	1. Alphanumeric POS keyboard MSR	PS/2, RS-485, USB
	2. Alphanumeric POS Keyboard keylock with mouse pointer	PS/2
	3. 50-key keyboard MSR	RS-485, USB
	4. 133-key keyboard MSR	RS-485, USB
	5. 4820 MSR	EIA-232, RS-485, USB
	6. 4840 MSR	EIA-232
	7. SureOne Built-In	PS/2
	8. CANPOS keyboard MSR (requires manual firmware update for support)	PS/2
	Kiosk	EIA-232
JUCC	1. 4820 MSR	EIA-232, RS-485, USB
	2. 4840 MSR	EIA-232
	3. Keyboard V MSR	RS-485, USB
	4. ANKPOS keyboard	PS/2, RS-485, USB
	5. 4685-KC1 (OPOS only)	RS-485
	6. 4685-K01 (OPOS only)	RS-485
	7. 4685-K02 (Ultra 7)	RS-485, USB
	8. 4674 Built-in	RS-485
	9. Kiosk	EIA-232

Table 118. MSR common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	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 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	All	All
DataEventEnabled	All	All
DeviceControlDescription	All	All

Table 118. MSR common properties (continued)

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

Table 119. MSR specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AccountNumber	All	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
CapTransmitSentinels	False	All
DecodeData	All	All
ErrorReportingType	All	All
ExpirationDate	All	All
FirstName	All	All
MiddleInitial	All	All
ParseDecodeData	All	All
ServiceCode	All	All
Suffix	All	All
Surname	All	All
Title	All	All
Track1Data	All	All
Track1DiscretionaryData	All	All
Track2Data	All	All
Track2DiscretionaryData	All	All
Track3Data	All	All
Track4Data	All JUCC	All JUCC
TracksToRead	All	All

Table 119. MSR specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
TransmitSentinels	Not supported	Not supported

Table 120. MSR common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	All	All
clearOutput	Not supported	Not supported
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	RS-485 – 4685-K02 (Ultra 7)	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 121. MSR events

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

MotionSensor

Table 122. MotionSensor supported devices

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

Table 123. MotionSensor common properties

Property	JavaPOS and OPOS Gateway	OPOS
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_NONE	PR_NONE
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
PowerNotify	All	All
PowerState	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 124. MotionSensor specific properties

Property	JavaPOS and OPOS Gateway	OPOS
Motion	All	All
Timeout	All	All

Table 125. MotionSensor common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All

Table 125. MotionSensor common methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
claim	Always shareable	Always shareable
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	Always shareable	Always shareable
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 126. MotionSensor specific methods

Method	JavaPOS and OPOS Gateway	OPOS
waitForMotion	All	All

Table 127. MotionSensor events

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

POSKeyboard

Table 128. POSKeyboard supported devices

Device		Connectivity
SBCS	1. Alphanumeric POS keyboard	PS/2, RS-485, USB
	2. Alphanumeric POS Keyboard keylock with mouse pointer	PS/2
	3. 50-key keyboard	RS-485, USB
	4. 133-key keyboard	RS-485, USB
	5. 32-key keypad	RS-485, USB, PS/2
CANPOS keyboard (automatic update from UPOS 1.7.5)		PS/2
DBCS	1. POS keyboard V	RS-485, USB
	2. Alphanumeric POS Korea keyboard	PS/2, RS-485, USB
	3. PLU keyboard 17	RS-485, USB
	4. Retail POS keyboard	RS-485, USB
		RS-485, USB
	5. SureOne Built-In Keyboard (JavaPOS only)	PS/2
	6. 4674 Built-in Keyboard	RS-485
	7. 4685–K02 (Ultra7) keyboard	RS-485, USB
	8. 4685-KC1 (OPOS only)	RS-485
	9. 4685-K01 (OPOS only)	RS-485

Table 129. POSKeyboard common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	All	All
DataEventEnabled	All	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All

Table 129. POSKeyboard common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 130. POSKeyboard specific properties

Property	JavaPOS and OPOS Gateway	OPOS
CapKeyUp	All	All
EventTypes	All	All
POSKeyData	All	All
POSKeyEventType	All	All

Table 131. POSKeyboard common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	All	All
clearOutput	All	All
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	All	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 132. POSKeyboard events

Event	JavaPOS and OPOS Gateway	OPOS
DataEvent	All	All

Table 132. POSKeyboard events (continued)

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

POSPrinter

Table 133. POSPrinter supported devices

Device	Connectivity	Comments
1. 4610 printer family	EIA-232	Includes SST (TM6/TF6/TM7/TF7), TI1, TI2, TI3, TI4, TI5, TI8, TI9 models
2. 4610 printer family	RS-485	Includes SST (TM6/TF6/TM7/TF7), TI1, TI2, TI3, TI4, TI5, TI8, TI9 models
3. 4610 printer family	USB	Includes SST (TM6/TF6/TM7/TF7), TI1, TI2, TI3, TI4, TI5, TI8, TI9 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 printer (thermal)	EIA-232	
12. SureOne printer (double-head impact)	EIA-232	OPOS only
13. SureOne printer (A04/A05 impact)	EIA-232	OPOS only

Table 134. POSPrinter common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	PR_STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	Not supported	Not supported
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All

Table 134. POSPrinter common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	All	All
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 135. POSPrinter specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	All
CapCharacterSet	All - Values based on HW	All - Values based on HW
CapConcurrentJrnRec	4, 5, 6, 7, 8, 9	4, 5, 6, 7, 8
CapConcurrentJrnSlp	Not supported	Not supported
CapConcurrentPageMode	Not supported	Not supported
CapConcurrentRecSlp	1, 2, 3 (TI1-5)	1, 2, 3 (TI1-5)
CapCoverSensor	All	All except SureOne printers
CapMapCharacterSet	Not supported	Not supported
CapRecPageMode	1,2,3	1,2,3 true only for TI4/TI8/TI9, TG5/TG9
CapSlpPageMode	Not supported	Not supported
CapTransaction	All	All
CapJrnPresent	4, 5, 6, 7, 8, 9	4, 5, 6, 7, 8
CapJrn2Color	Not supported	Not supported
CapJrnBold	4, 5, 7, 8, 9	4, 5
CapJrnCartridgeSensor	Not supported	Not supported
CapJrnColor	Not supported	Not supported
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	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) EC > 0*x*33	1, 2, 3 (TI3-4,Tx6-8) EC > 0*x*33
CapRecBarCode	1, 2, 3, 7, 8, 9, 11	1, 2, 3, 7, 8, 11

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapRecBitmap	All	All
CapRecBold	1, 2, 3, 4, 5, 7, 8, 9, 11	1, 2, 3, 4, 5, 10, 11, 13
CapRecCartridgeSensor	Not supported	Not supported
CapRecColor	1, 2, 3 (TI3-4,Tx6-9) EC > 0*x*33	1, 2, 3 (TI3-4,Tx6-8) EC > 0*x*33
CapRecDhigh	1, 2, 3, 7, 8, 9, 11 4, 5 DH forces DW reverts to normal Rotation	1, 2, 3, 7, 8, 11, 13 4, 5 DH forces DW 10 reverts to normal rotation in 180 mode
CapRecDwide	1, 2, 3, 4, 5, 7, 8, 9, 11	1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13
CapRecDwideDhigh	1, 2, 3, 4, 5, 7, 8, 9, 11	1, 2, 3, 4, 5, 7, 8, 11, 13 10 reverts to normal rotation in 180 mode
CapRecEmptySensor	1, 2, 3, 6, 7, 8, 9	1, 2, 3, 6
CapRecItalic	Not supported	Not supported
CapRecLeft90	1, 3, EIA-232 USB (Linux)	Not supported
CapRecMarkFeed	Not supported	Not supported
CapRecNearEndSensor	7, 8	7
CapRecPapercut	All	All except 10, 12, and 13
CapRecRight90	1, 3, EIA-232 USB (Linux) 8, 9	1, 2, 3, 8,
CapRecRotate180	1, 2, 3, 11 EC > 0*x*33	1, 2, 3 EC > 0*x*33 & 10, 11, 12, 13
CapRecStamp	1, 2, 3, 11- uses stored bitmap 1 6 - physical stamp 7, 8, 9- downloaded stamp	1, 2, 3 - uses stored bitmap 1 6 - physical stamp 7, 8, 9- downloaded stamp
CapRecUnderline	1, 2, 3, 7, 8, 9, 11	1, 2, 3, 7, 8, 10, 11, 12, 13
CapSlpPresent	1, 2, 3, 4, 5	1, 2, 3, 4, 5
CapSlpFullslip	4, 5	4, 5
CapSlp2Color	Not supported	Not supported
CapSlpBarCode	1, 2, 3 EC > 0*x*1D	1, 2, 3 EC > 0*x*1D
CapSlpBitmap	1, 2, 3, 4, 5	1, 2, 3, 4, 5, 6
CapSlpBold	1, 2, 3, 4, 5	1, 2, 3, 4, 5
CapSlpBothSidesPrint	1, 2, 3 except SST	1, 2, 3 except SST
CapSlpCartridgeSensor	Not supported	Not supported
CapSlpColor	Not supported	Not supported
CapSlpDhigh	1, 2, 3 4, 5 DH forces DW	1, 2, 3 4, 5 DH forces DW
CapSlpDwide	1, 2, 3, 4, 5	1, 2, 3, 4, 5
CapSlpDwideDhigh	1, 2, 3, 4, 5	1, 2, 3, 4, 5
CapSlpEmptySensor	1, 2, 3, 4, 5	1, 2, 3, 4, 5
CapSlpItalic	Not supported	Not supported
CapSlpLeft90	1, 2, 3	1, 2, 3 (except TI5 and Tx7)

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
CapSlpNearEndSensor	1, 2, 3, 4, 5	1, 2, 3, 4, 5, 6
CapSlpRight90	Not supported	Not supported
CapSlpRotate180	Not supported	Not supported
CapSlpUnderline	Not supported	Not supported
CartridgeNotify	Not supported	Not supported
CharacterSet	All	All
CharacterSetList	1,2,3 - "437, 858, 860, 863, 865, 998" 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: 936	All
CoverOpen	1, 2, 3, 4, 5, 11 7, 8, 9 - when printer not idle	1, 2, 3, 4, 5 7, 8 - when printer not idle
ErrorLevel	All	All
ErrorStation	All	All
ErrorString	All	All
FlagWhenIdle	All	All
FontTypefaceList	Not supported	1, 2, 3 (USB Windows) (TI3-5, Tx6-8) EC > 0*x*33 can list ″Fixed, Proportional″ based on downloaded UD Fonts, null string otherwise
JrnCartridgeState	Not supported	Not supported
JrnCurrentCartridge	Not supported	Not supported
JrnEmpty	4, 5, 6, 7, 8. 9	4,5,6,7,8
JrnLetterQuality	Not supported	Not supported
JrnLineChars	1, 2, 3 - 0 4, 5 - 38 6 - 25 7, 8, 9 - 32	1, 2, 3, 10, 11, 12, 13 - 0 4, 5 - 38 6 - 25 7, 8 - 32

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
JrnLineCharsList	1, 2, 3 - ″″ (empty) 4, 5 - ″30,38″ 6 - ″25,30″ 7, - ″32,42″ 8, 9 - "32"	1, 2, 3, 10, 11, 12, 13 - ″″ (empty) 4, 5 - ″30,38″ 6 - ″25,30″ 7, 8- ″32,42″
JrnLineHeight	1, 2, 3 - 0 4, 5, 6 - 9 7, 8, 9 - 24	1, 2, 3, 10, 11, 12, 13 - 0 4, 5, 6 - 9 7, 8- 24
JrnLineSpacing	1, 2, 3 - 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 - 0 4, 5 - 380 6 - 300 7, - 420 8, 9 - 432	1, 2, 3, 10, 11, 12, 13 - 0 4, 5- 380 6 - 300 7, 8- 420
JrnNearEnd	7, 8, 9	7, 8
MapCharacterSet	Not supported	Not supported
MapMode	All	All
PageModeArea	4610: When PageModeStation is set ″0,0,576,1250 (for TI8/9) or ″0,0,576,800″; else zero	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.
		Zero

PageModeHorizontalPosition	4610: Zero before setting PageModeStation	Zero
PageModePrintArea	4610: When PageModeStation is set, max is PageArea value, else zero	Zero
PageModePrintDirection	4610: Initialized to jpos.POSPrinterConst.PTR_PD_LEFT_TO_RIGHT when the device is first enabled	Zero

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
PageModeStation	4610: jpos.POSPrinterConst.PTR_S_RECEIPT	Zero
PageModeVerticalPosition	4610: Zero before setting PageModeStation	Zero
RecBarCodeRotationList	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 can list ″0,180″. Otherwise, 1, 2, 3 list ″0″. 8, 9, - "0"	1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33 can list ″0,180″. Otherwise, 1, 2, 3 list ″0″.
RecBitmapRotationList	″0″	″0″
RecCartridgeState	Not supported	Not supported
RecCurrentCartridge	Not supported	Not supported
RecEmpty	1, 2, 3, 6, 7, 8, 9	1, 2, 3, 6
RecLetterQuality	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33	1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33
RecLineChars	1, 2, 3 - 44 4, 5, 6, 7, - 38 11 - 48 TI5, TM7 - EIA-232 and USB (Linux) – 32 TI5, TM7 – 38 8, 9 - 32	1, 2, 3 (TI3-5, Tx6-8 EC > 0*x*33) - 56 1, 2, 3 (all other) - 48 4, 5, 6, 7, 8 - 38 10, 11, 12, 13 – 40
RecLineCharsList	1, 2, 3 (TI3-5, Tx6-7) EC > 0*x*33 - ″34,44,48,57″ 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″ 1, 2, 3 (EIA-232 and USB Linux) – ″34,44,48,57″ TI5, TM7 – ″38,44,48″ TI5 USB (Linux), EIA-232 – ″32″	1 (TI3-5, Tx6-8) EC > 0*x*33 - ″34,44,48,57″ 2, 3 (TI3-5, Tx6-8) EC > 0*x*33 - ″34,44,48,58″ 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 - 20 4, 5, 6 - 9 7, 8, 9 - 24 11 - 24	1 - 20 2, 3 - 34 4, 5, 6 - 9 7, 8, 11- 24 10, 12, 13 - 8

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
RecLineSpacing	1, 2, 3 - 34 4, 5, 6 - 12 7, - 27 8, 9, - 30 11 - 32	1, 2, 3 - 34 4, 5, 6 - 12 7, 8- 27 10, 12, 13 - 9 11 - 32
RecLinesToPaperCut	1, 2, 3 - 12 4, 5, 7, - 6 8, 9 - 3 6 - 16 11 - 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	1, 2, 3 (TI1-5, Tx6-8) - 576 1, 2, 3 Tx6-8 narrow paper - 400 4, 5 - 380 6 - 300 7, 8 - 420 10, 12, 13 - 280 11 - 576
RecNearEnd	7, 8	7
RecSidewaysMaxLines	1, 2, 3 TI1-5, Tx6-9 - 17 1, 2, 3 Tx6-9 Narrow Paper - 12 7, 8, 9 - 14	1 TI1-5, Tx6-8 - 16 2, 3 TI1-5, Tx6-8 - 14 1, 2, 3 Tx6-8 Narrow Paper - 18 7, 8 - 19
RecSidewaysMaxChars	1, 2, 3 - 61 7, 8, 9 - 256	1 - 61 2, 3, 7, 8 - 53
RotateSpecial	_NORMAL: 1, 2, 3, 8, 9, 11 _LEFT90: — _RIGHT90: — 8, 9 _ROTATE180: 1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33	_NORMAL: 1, 2, 3 _LEFT90: — _RIGHT90: — _ROTATE180: 1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33
SlpBarCodeRotationList	″0″	1, 2, 3 - (TI3-5, Tx6-8) EC > 1D, has ″0″
SlpBitmapRotationList	″0″	″0″
SlpCartridgeState	Not supported	Not supported
SlpCurrentCartridge	Not supported	Not supported
SlpEmpty	1, 2, 3, 4, 5	1, 2, 3, 4, 5

Table 135. POSPrinter specific properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
SlpLetterQuality	1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33	1, 2, 3 (TI3-4, Tx6-8, TI8, TI9) EC > 0*x*33 (not supported on TI5)
SlpLineChars	1, 2, 3 - 47 4, 5 - 86 TI5 - 19 TI5 USB (Linux) and EIA-232 – 21	1, 2, 3 - 47 4, 5 - 86 6 - 58
SlpLineCharsList	1, 2, 3 - ″37,47,52″ 4, 5 - ″30,38,42,68,86,94″ — older models only support 30, 38, 68, 86 TI5 - ″18, 19″ TI5 USB (Linux) and EIA-232 – ″21″ 1, 3 USB (Linux) and EIA-232 – ″27, 47, 52″	1 - ″37,47,52″ 2, 3 - ″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 - 7 4, 5 - 9	1 - 4 2, 3 - 7 4, 5, 6 - 9
SlpLinesNearEndToEnd	All - values based on HW	All - values based on HW
SlpLineSpacing	1, 2, 3 - 8 4, 5 - 12	1 - 9 2, 3 - 8 4, 5, 6 - 12
SlpLineWidth	1, 2, 3 - 470 4, 5 - 880	1 - 474 2, 3 - 470 4, 5 - 880 6 - 300
SlpMaxLines	All - values based on HW	All - values based on HW
SlpNearEnd	1, 2, 3, 4, 5	1, 2, 3, 4, 5, 6
SlpPrintSide	1, 2, 3 except SST	1, 2, 3 except SST
SlpSidewaysMaxLines	1, 2, 3 - 19	1 - 18 2, 3 - 19
SlpSidewaysMaxChars	1, 2, 3 - 148	1, 2, 3 - 147

Table 136. POSPrinter common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	INTERNAL = ALL EXTERNAL = ALL INTERACTIVE = Not supported	All
claim	All	All

Table 136. POSPrinter common methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
clearInput	Not supported	Not supported
clearOutput	All	All
close	All	All
compareFirmwareVersion	Not supported	Not supported
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	All
release	All	All
resetStatistics	Not supported	Not supported

Table 136. POSPrinter common methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
retrieveStatistics	True (see Appendix A, "JavaPOS support for UnifiedPOS device statistics properties," on page 341)	Supported: 4610 and SureOne
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 137. POSPrinter 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	1, 2, 3 except SST and TI5
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	Not supported
pageModePrint	4610: Control is one of the following: PTR_PM_PAGE_MODE, PTR_PM_PRINT_SAVE, PTR_PM_NORMAL, PTR_PM_CANCEL	Not supported
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, Tx6-8) EC>1D (slip)
printBitmap	All 8, 9 (receipt)	All except 9
printImmediate	All	All
printNormal	All	All
printTwoNormal	All except 11	1, 2, 3, 4, 5, 6, 7, 8
rotatePrint	_NORMAL: All _LEFT90: 1, 2, 3 (slip), 1,2,3 (receipt) - USB (Linux) and EIA-232 _RIGHT90: 1, 2, 3 (receipt) - USB (Linux) and EIA-232, 7,8,9 (receipt) _ROTATE180: 1, 2, 3, 11 (TI3-5, Tx6-9) EC > 0*x*33 (receipt)	_NORMAL: All _LEFT90: 1, 2, 3 (slip) _RIGHT90: 1, 2, 3, 7, 8 (receipt) _ROTATE180: 1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33 (receipt), 10, 11, 12, 13 (receipt)
setBitmap	1, 2, 3, 4, 5, 7, 8, 9, 11	All except 9
setLogo	All	All
transactionPrint	All	All
validateData	All	All

Table 138. POSPrinter events

Event	JavaPOS and OPOS Gateway	OPOS
DirectIOEvent	All	1, 2, 3

Table 138. POSPrinter events (continued)

Event	JavaPOS and OPOS Gateway	OPOS
ErrorEvent	All	All
OutputCompleteEvent	All	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	Not supported
PTR_SUE_REC_COVER_OPEN	1,2,3,4,5,11	Not supported
PTR_SUE_SLP_COVER_OK	1,2,3 except SST	Not supported
PTR_SUE_SLP_COVER_OPEN	1,2,3 except SST	Not supported
PTR_SUE_JRN_EMPTY	4,5,6,7,8	4,5,6,7,8
PTR_SUE_JRN_NEAREMPTY	7,8,9	7,8,9
PTR_SUE_JRN_PAPEROK	4,5,6,7,8,9	4,5,6,7,8,9
PTR_SUE_REC_EMPTY	1,2,3,6,7,8,9	1,2,3,6,7,8,9
PTR_SUE_REC_NEAREMPTY	7,8	7,8
PTR_SUE_REC_PAPEROK	1,2,3,6,7,8,9	1,2,3,6,7,8,9
PTR_SUE_SLP_EMPTY	1,2,3,4,5	1,2,3,4,5
PTR_SUE_SLP_NEAREMPTY	1,2,3,4,5	1,2,3,4,5
PTR_SUE_SLP_PAPEROK	1,2,3,4,5	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
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

To determine if escape sequences or data can be performed on a printer station, the application can call the *validateData* method. (For some escape sequences, corresponding capability properties can also be used.) Escape sequences are assumed to be supported on all stations for the printer specified. Otherwise, only stations that support the escape sequence are indicated in parenthesis (). Escape sequences that are suported in a given station could have unexpected results.

Table 139. POSPrinter escape sequences

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
Alternate color	ESC |rC	1, 2, 3 (TI3-5, Tx6-7) EC > 0x33 with Configured setting and correct paper (receipt)	1, 2, 3 (TI3-5, Tx6-7) EC > 0*x*33 with Configured setting and correct paper
Alternate color (Custom)	ESC |#rC	1, 2, 3 (TI3-5, Tx6-7) EC > 0*x*33 with Configured setting and correct paper (receipt)	1, 2, 3 (TI3-5, Tx6-7) EC > 0*x*33 with Configured setting and correct paper

Table 139. POSPrinter escape sequences (continued)

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
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	All
Double high	ESC |3C	Same as CapXxxDhigh	Same as CapXxxDhigh
Double high & wide	ESC |4C	Same as CapXxxDwideDhigh	Same as CapXxxDwideDhigh
Double wide	ESC |2C	Same as CapXxxDwide	Same as CapXxxDwide
Feed and paper cut	ESC |#fP	All (receipt)	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	All
Feed reverse	ESC |#rF	4, 5, 8, 11 - Cannot mix with feed forward (receipt)	4, 5 - Cannot mix with feed forward
Feed units	ESC |#uF	All	All
Fire stamp	ESC |sL	• 1, 2, 3, 11 - 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, Tx6-8) EC > 0*x*33 with downloaded proportional font
Integrated escape	ESC |#E	Not supported	Not supported
Italic	ESC |iC	Not supported	Not supported
Normal	ESC |N	All	All
Paper cut	ESC |#P	All (receipt)	All
Print bitmap	ESC |#B	• 1, 2, 3, 4, 5, 11 • 7, 8, 9 (receipt)	1, 2, 3, 4, 5, 7, 8, 10, 11, 12
Print bottom logo	ESC |bL	All	All
Print top logo	ESC |tL	All	All
Reverse video	ESC |rvC	• 1, 2, 3 (receipt) • 7, 8, 9	1, 2, 3, 8, 11 (11 reverts to normal rotation in 180 mode)
RGB color	ESC |#fC	Not supported	Not supported
Right justify	ESC |rA	All	All

Table 139. POSPrinter escape sequences (continued)

Operation	Escape sequence	JavaPOS and OPOS Gateway	OPOS
Scale horizontally	ESC |#hC	1, 2, 3, 4, 5, 6, 7, 8, 9. Up to 2. Same as CapXxxDwide 1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 Up to 8 (receipt) 11: Up to 6 (receipt)	1, 2, 3, 4, 5, 6, 7, 8, 11. Up to 2. Same as CapXxxDwide 1, 2, 3 (TI3-5, Tx6-8) EC > 0*x*33 Up to 8
Scale vertically	ESC |#vC	1, 2, 3, 4, 5, 6, 7, 8, 9. Up to 2 is for slip; up to 8 is for receipt. Same as CapXxxDhigh 1, 2, 3 (TI3-5, Tx6-9) EC > 0*x*33 Up to 8 (receipt) 11: Up to 6 (receipt)	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 (TI3-5, Tx6-8) EC > 0*x*33 Up to 8
Shading	ESC |#sC	7	7, 8
Single high & wide	ESC |1C	All	All
Subscript	ESC |tbC	Not supported	Not supported
Superscript	ESC |tpC	Not supported	Not supported
Underline	ESC |#uC	Same as CapXxxUnderline	Same as CapXxxUnderline

Table 140. POSPrinter common statistics

XML definition name	4610	4689
DeviceCategory	Not supported	Not supported
FirmwareRevision	Not supported	Not supported
Installationdate	Not supported	Not supported
Interface	Not supported	Not supported
ManufactureDate	Not supported	Not supported
ManufacturerName	Not supported	Not supported
ModelName	Not supported	Not supported
SerialNumber	Not supported	Not supported
UnifiedPOSVersion	Not supported	Not supported

Table 141. POSPrinter specific statistics

XML definition name	4610	4689
BarcodePrintedCount	Not supported	Not supported
FailedPaperCutCount	Not supported	Not supported
FailedPrintSideChangeCount	Not supported	Not supported
FormInsertionCount	Not supported	Not supported
HomeErrorCount	Not supported	Not supported
JournalCharacterPrintedCount	Not supported	Not supported
JournalLinePrintedCount	Not supported	Not supported
MaximumTempReachedCount	Not supported	Not supported

Table 141. POSPrinter specific statistics (continued)

XML definition name	4610	4689
NVRAMWriteCount	Not supported	Not supported
PaperCutCount	Not supported	Not supported
PrinterFaultCount	Not supported	Not supported
PrintSideChangeCount	Not supported	Not supported
ReceiptCharacterPrintedCount	Not supported	Not supported
ReceiptCoverOpenCount	Not supported	Not supported
ReceiptLineFeedCount	Not supported	Not supported
ReceiptLinePrintedCount	Not supported	Not supported
SlipCharacterPrintedCount	Not supported	Not supported
SlipCoverOpenCount	Not supported	Not supported
SlipLineFeedCount	Not supported	Not supported
SlipLinePrintedCount	Not supported	Not supported
StampFiredCount	Not supported	Not supported

Scale

Table 142. Scale supported devices

Device	Connectivity	Comments
1. IBM 4687 Compatible Scanner/scale	RS-485	
	USB	Using ProtocolConverter
2. IBM 4696 Compatible Scanner/scale	RS-485	
	USB	Using ProtocolConverter
3. IBM 4698 Compatible Scanner/scale	RS-485	
	USB	Using ProtocolConverter
4. OEM Scale	USB	Must conform to *IBM USB OEM Point-of-Sale Device Interface Specification* V1.0 or later

Table 143. Scale common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	All	All
DataEventEnabled	All	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All

Table 143. Scale common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
State	All	All

Table 144. Scale specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	All
CapDisplay	All	All
CapDisplayText	Not supported	Not supported
CapPriceCalculating	Not supported	Not supported
CapStatusUpdate	False	False
CapTareWeight	Not supported	Not supported
CapZeroScale	4	All except 1
MaxDisplayTextChars	Not supported - 0	Not supported - 0
MaximumWeight	All	All
SalesPrice	Not supported	Not supported
ScaleLiveWeight	Not supported	Not supported
StatusNotify	Not supported	Not supported
TareWeight	Not supported	Not supported
UnitPrice	Not supported	Not supported
WeightUnit	All	All

Table 145. Scale common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	All	All
clearOutput	All	All
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 146. Scale specific methods

Method	JavaPOS and OPOS Gateway	OPOS
displayText	Not supported	Not supported

Table 146. Scale specific methods (continued)

Method	JavaPOS and OPOS Gateway	OPOS
readWeight	All	All
zeroScale	4	All except 1

Table 147. Scale events

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

Scanner (bar code reader)

Table 148. 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

Table 149. Scanner common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	All	All
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
DataCount	All	All
DataEventEnabled	All	All
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All

Table 149. Scanner common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
FreezeEvents	All	All
OpenResult	Not supported	All
OutputID	Not supported	Not supported
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 150. Scanner specific properties

Property	JavaPOS and OPOS Gateway	OPOS
DecodeData	All	All
ScanData	All	All
ScanDataLabel	All	All
ScanDataType	All	All

Table 151. Scanner common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearInput	All	All
clearOutput	All	All
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 152. Scanner events

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

Table 152. Scanner events (continued)

Event	JavaPOS and OPOS Gateway	OPOS
StatusUpdateEvent	All	All

ToneIndicator

Table 153. ToneIndicator supported devices

Device		Connectivity
SBCS keyboard	1. Alphanumeric POS keyboard tone	PS/2, RS-485, USB
	2. Alphanumeric POS keyboard tone with mouse pointer	PS/2
	3. 50-key keyboard tone	RS-485, USB
	4. 133-key Keyboard tone	RS-485, USB
	5. 4820 tone	RS-485, USB, PS/2
DBCS keyboard	1. POS Keyboard V tone	RS-485, USB
	2. Alphanumeric POS Korea keyboard tone	PS/2, RS-485, USB
	3. PLU keyboard tone	RS-485, USB
	4. PC POS keyboard	RS-485, USB
	5. 4685-KC1	RS-485 (OPOS only)
	6. 4685-K01	RS-485 (OPOS only)
	7. 4685-K02 (Ultra 7)	RS-485, USB
Printer	4610 SST Tx6/Tx7	EIA-232, RS-485, USB
System tone	Internal speaker	Integrated (OPOS only)

Table 154. ToneIndicator common properties

Property	JavaPOS and OPOS Gateway	OPOS
AutoDisable	Not supported	Not supported
BinaryConversion	Not supported	All
CapCompareFirmwareVersion	False	False
CapPowerReporting	PR_STANDARD	All support STANDARD
CapStatisticsReporting	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CapUpdateFirmware	False	False
CapUpdateStatistics	False (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	False
CheckHealthText	All	All
Claimed	All	All
ClearInput	Not supported	Not supported
DataCount	Not supported	Not supported
DataEventEnabled	All	Not supported
DeviceControlDescription	All	All
DeviceControlVersion	All	All
DeviceEnabled	All	All
DeviceServiceDescription	All	All
DeviceServiceVersion	All	All
FreezeEvents	All	All

Table 154. ToneIndicator common properties (continued)

Property	JavaPOS and OPOS Gateway	OPOS
OpenResult	Not supported	All
OutputID	All	All
PowerNotify	All	All
PowerState	All	All
PhysicalDeviceDescription	All	All
PhysicalDeviceName	All	All
ResultCode	Not supported	All
ResultCodeExtended	Not supported	All
State	All	All

Table 155. ToneIndicator specific properties

Property	JavaPOS and OPOS Gateway	OPOS
AsyncMode	All	All
CapPitch	All	All
CapVolume	All	All
InterToneWait	All	All
Tone1Duration	All	All
Tone1Pitch	All	All
Tone1Volume	All	All
Tone2Duration	All	All
Tone2Pitch	All	All
Tone2Volume	All	All

Table 156. ToneIndicator common methods

Method	JavaPOS and OPOS Gateway	OPOS
checkHealth	All	All
claim	All	All
clearOutput	All	All
close	All	All
compareFirmwareVersion	Not supported	Not supported
directIO	Not supported	Not supported
open	All	All
release	All	All
resetStatistics	Not supported	Not supported
retrieveStatistics	True (see Appendix A, “JavaPOS support for UnifiedPOS device statistics properties,” on page 341)	Not supported
updateFirmware	Not supported	Not supported
updateStatistics	Not supported	Not supported

Table 157. ToneIndicator specific methods

Method	JavaPOS and OPOS Gateway	OPOS
sound	All	All
soundImmediate	All	All

Table 158. ToneIndicator events

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

Chapter 6. Problem determination

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

JavaPOS problems

Tracing

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 MaxbackIndex is greater than 1, then the log file is rotated.

Logging

Logging is stored according to the device that produced it. For example, logging for MSR is stored in a file named MSR.log, for MICR it is MICR.log, and so on. If the message being logged is not device specific, it is stored in a file named AllDevices.log instead of <device>.log, where <device> = MSR, MICR, scanner, and so on.

The properties file for logging can be created and accessed as follows:

1. Start the POS Control Center in one of two ways:
   a. GUI: **Programs** > **IBM JavaPOS** > **POS Control Center**
   b. Command line: `java com.ibm.jpos.tools.sdicc.ControlCenterApp`
2. Select **File** > **Properties**.
3. Select the log4jConfiguration tab.
4. Modify applicable properties.

The location for these files is <HOME>/.ibmjpos/logging/; where <HOME> is the absolute path to the user's home directory. The root.log file (in the same path that stores the logging for all of the devices) will contain all the messages logged, regardless if they were logged for a specific device or for AllDevices category.

To enable or disable logging, you must change the <HOME>/.ibmjpos/config/log4jConfiguration.properties file. If you want to disable logging for a specific category, or change the level of logging at the definition of the category, change this file. The following example shows how to disable logging for MSR:

```
change:  log4j.logger.MSR=ALL, MSRAppender
    to:  log4j.logger.MSR=FATAL, MSRAppender
```

If you want to disable logging for all the categories, change the level of logging to FATAL at the definition of the rootCategory:

```
 change:  log4j.rootLogger=ALL, rootAppender
     to:  log4j.rootLogger=FATAL, rootAppender
```

Details of levels and categories hierarchy are found in "Logging details" on page 213. If categorized logging for other devices is desired, add the corresponding entries for those devices to the file mentioned before in the following way:

1. First set the new category, its level and appenders. For example: `log4j.logger.MyLogger=ALL, MyAppender`. The type of the appenders must be defined.
2. Next, set the appenders type. For example: `log4j.appender.MyAppender=org.apache.log4j.RollingFileAppender` or `log4j.appender.MyAppender=org.apache.log4j.ConsoleAppender`
3. Set the file name for the FileAppenders. Notice that absolute paths are required in the value of this property. For example: `log4j.appender.MyAppender.File=/home/user/.ibmjpos/logging /MyLogFile.log`. The file path can include a reference to the home path as follows: `log4j.appender.MSRAppender.File=<home>/.ibmjpos/logging/MSR.log`.

   The maximum file size is set using the following: `log4j.appender.MSRAppender.MaxFileSize=1000`. The length is specified in bytes.
4. Set the specific layout that each appender is going to use. For example, if PatternLayout is used, the conversionPattern determines the way in which the data will be displayed. The conversion pattern is closely related to the conversion pattern of the printf function in C language. A conversion pattern is composed of literal text and format control expressions called conversion specifiers. You may insert any literal text within the conversion pattern. Each conversion specifier starts with a percent sign (%) and is followed by optional format modifiers and a conversion character.

   The conversion character specifies the type of data. For example; category, priority, date, thread name. The format modifiers control such things as field width, padding, left and right justification. See

   `http://logging.apache.org/log4j/docs/api/org/apache/log4j/PatternLayout.html`

   for the meaning of each conversion specifier.

   See the following example:

   ```
   log4j.appender.MyAppender.layout=org.apache.log4j.PatternLayout
   log4j.appender.MyAppender.layout.ConversionPattern=%d %-5p %c - %m%n
   ```

Logging details

The levels of logging are as follows in decreasing order of detail: ALL, DEBUG, INFO, WARN, ERROR, FATAL and OFF. The logging is categorized to assign a level to each category. If a category has a level equal to or less than the level that you are using to log, then that log entry is stored, otherwise the logEntry is ignored.

The next rule is quoted from the log4j manual (`logging.apache.org/log4j/docs/manual.html`)

> A logging request is said to be enabled if its level is higher than or equal to the level of its Logger. Otherwise, the request is said to be disabled. A Logger without an assigned level inherits one from the hierarchy.

The Loggers are in a hierarchy, so if a specific logger does not have a level assigned, it inherits the level of its ancestor. A rootCategory always exists, so if you set the level of the rootCategory to FATAL, the logging is OFF regardless of the level of its children.

Logging output

JavaPOS drivers define settings for the log4j rootLogger. This causes all logging statements from JavaPOS and the application logging to be sent to the same output file, called *root.log*. The location of this file is defined by the following property:

- File name: Log.properties
- Location:
  - Windows: <pos install dir>\ibmjpos\config
  - Linux: /opt/ibm/javapos/etc
- Property: com.ibm.jutil.logging.default.Log4jLogStore.LogFilePath=<home>/.ibmjpos/logging

How to disable JavaPOS defined logging

In case of conflicts, you can disable JavaPOS logging by commenting out the following property:

- File name: log4jConfiguration.properties
- Location:
  - Windows: <pos install dir>\ibmjpos\config
  - Linux: /opt/ibm/javapos/etc
- Property: log4j.rootLogger=ALL, rootAppender *(comment out this property)*

If the application has defined log4j.rootLogger property, and JavaPOS logging has been disabled, then the logging output is redirected to the defined application output.

Javax.usb for Windows before UPOS 1.9.2

Prior to UPOS 1.9.2, Javax.usb was divided into three parts, each with specific considerations:

- DLL layer tracing
- kernel driver tracing
-

DLL layer tracing

This tracing is directed to the console. To enable tracing, at least one of the properties in javaxusb.properties must be set to "true".

basic tracing
com.ibm.jusb.os.windows.WindowsUsbServices.trace_data = false

trace statistics from DLL
com.ibm.jusb.os.windows.WindowsUsbServices.native_trace_statistic = false

trace communication packets (hex dump)
com.ibm.jusb.os.windows.WindowsUsbServices.native_trace_packet = false

trace general information from the DLL
com.ibm.jusb.os.windows.WindowsUsbServices.native_trace_info = false

trace errors from the DLL
com.ibm.jusb.os.windows.WindowsUsbServices.native_trace_error = false

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.

Javax.usb for Windows with UPOS 1.9.2 or later

Release 1.9.2 or UPOS introduces 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:

- `com.ibm.jusb.util.UsbTracer=com.ibm.jusb.os.windows.WindowsStdOutUsbTracer`

  This property specifies the tracer to use. In this example, tracing is directed to the console (default) by "WindowsStdOutUsbTracer".
- `com.ibm.jusb.os.windows.WindowsUsbServices.nativetracelevel=TRACE_CRITICAL`

  This property sets the DEBUG level for the DLL tracing.
- `com.ibm.jusb.util.WindowsUsbTracer.currentLevel=TRACE_OFF`

  This property sets the DEBUG level for the javax.usb for Windows layer. Setting this property to TRACE_OFF disables javax.usb for Windows tracing.

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.util.UsbTracer=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 oncorporated 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.

Javax.usb for Linux

The Linux implementation 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`

  This either disables tracing entirely, or 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. Note that if append mode is used, the trace file will grow ever-larger every time javax.usb is run, and 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 information

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

1. From a command line prompt, type in AIPTRACED. The application will appear minimized in the task list.

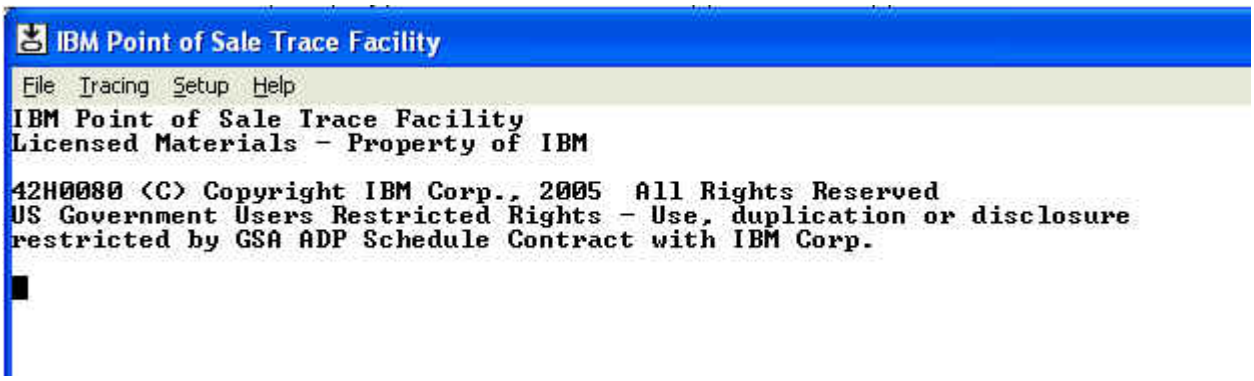


Figure 30. POS tracing facility

2. Select Setup and then tracing to choose item requested for tracing. In the setup window, you can set maximum trace file size, trace file name, backup Count (the number of backup trace files to save), startup Mode, file mode, and visibility.

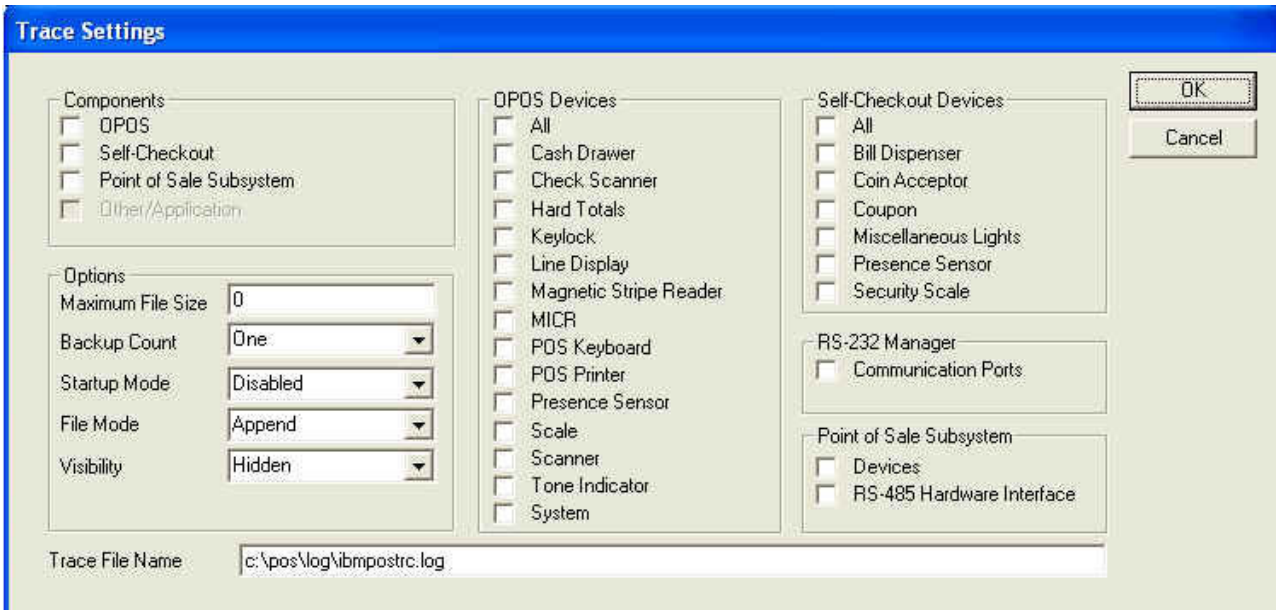


Figure 31. POS tracing facility

3. When complete, click **OK** and select Tracing and enable.

For the OPOS Gateway, perform one of the following two sets of instructions:

-
  1. From a command line prompt, change to the directory where your application resides, or the setting can be put in the System Environment variables section of the control panel.
  2. Enter **set AIP_OPOS_TRACE=ON**
  3. Enter **set AIP_OPOS_TRACE_ALL=ON**
  4. Start your application from the command line prompt.
- or
  1. Find the entry for the specific device for which trace is desired under the OleForRetail key.
  2. Add a value of type DWORD to the entry called TraceFlags
  3. Set TraceFlags to a value of 1 to 4, 1 being the highest level, such as errors and warning, 4 being the lowest level, showing all method calls.

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.

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

Chapter 7. Modifying Service Object behavior for OPOS

Service Object behavior can and should be modified using the Configuration tool. In nearly all cases, this is all that is required. In some special cases additional settings are available in the registry to further modify Service Object Behavior. This chapter lists the settings and their uses.

All devices

Table 159. Service Object settings for all devices

Keyword	Type	Description
deviceName	String	Name of device (1-10 characters)

RS-485/USB devices

General settings

Table 160. Service Object settings for RS-485/USB devices (general)

Keyword	Type	Description
slotNumber	String	Slot (0-8) to which the device is connected.
portNumber	String	Port to which the device is connected. Valid values are: 0, X'11', X'22'
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* (SC30-3560).

Cash drawer

Table 161. Service Object settings for RS-485/USB devices (cash drawer)

Keyword	Type	Description
CashDrawerNumber	String	Cash drawer number. Valid values are 1 (default) and 2.
SignalsReversed	String	Whether reverse wiring polarity is used for cash drawer. Valid values are: **True** Polarity is reversed **False** Polarity is not reversed (default)

Hard totals

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

Table 162. Service Object settings for RS-485/USB devices (hard totals)

Keyword	Type	Description
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)
HarddiskFileName	String	Path name of file to use for Hard Totals storage on hard disk.
HarddiskFileSize	String	Size (in bytes) of file used for Hard Totals Storage on hard disk.

Line display

Table 163. Service Object settings for RS-485/USB devices (line display)

Keyword	Type	Description
DefineCharacter	Key	Key values for user-defined characters. **Note:** This setting cannot be modified with the Configuration tool.
DefineCharacter\<X>	String	Decimal value of user-defined character to be downloaded. Eight bytes of binary data for character definition. For character format, refer to *IBM Point of Sale Subsystem Programming Reference and User's Guide* (SC30-3560). **Note:** This setting cannot be modified with the Configuration tool.

Keylock

Table 164. Service Object settings for RS-485/USB devices (keylock)

Keyword	Type	Description
SecondKeyPosition	String	When present, the additional position on the Japanese ANPOS keyboard is mapped to a LOCK_KP_LOCK.
PositionCount	DWORD	4685-K02 only. Number of keylock positions (1-6).
PositionMapFrom	Binary	4685-K02 only. Keylock position code mapping (from).
PositionMapTo	Binary	4685-K02 only. Keylock position code mapping (to).
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.

Table 164. Service Object settings for RS-485/USB devices (keylock) (continued)

Keyword	Type	Description
OnlineTimeout	String	Time in milliseconds to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.

MICR

Table 165. Service Object settings for RS-485/USB devices (MICR)

Keyword	Type	Description
ExceptionFile	String	Fully-qualified filename of MICR exception file, which contains exception processing information. The default value is: C:\POS\BIN\PARSE.DAT
MinMicrSignalLevel	String	Minimum signal string (0-100) for successful MICR read. The default value is 50 (full string).
SlipRegTimeout	String	Time (in milliseconds) to wait for check to be registered. **Note:** This setting cannot be modified with the Configuration tool.
SlipRegMinDelay	String	Time (in milliseconds) to delay before sending document register command. **Note:** This setting cannot be modified with the Configuration tool.
StrictMICRParsing	String	Whether strict parsing of MICR data is used. Valid values are: **True** Strict parsing **False** Normal parsing (default) **Note:** This setting cannot be modified with the Configuration tool.

POS keyboard

Table 166. Service Object settings for RS-485/USB devices (POS keyboard)

Keyword	Type	Description
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.
NumlockOn	String	Initial state of Num Lock light. Overrides the Numlock setting. Valid values are: **True** Illuminated **(other value)** Dark (default)
NumlockEnabled	String	Whether the Num Lock key is enabled. Valid values are: **True** Enabled **(other value)** Disabled (default)
ScrolllockOn	String	Initial state of Scroll Lock light. Valid values are: **True** Illuminated **(other value)** Dark (default)
ScrolllockEnabled	String	Whether the Scroll Lock key is enabled. Valid values are: **True** Enabled **(other value)** Disabled (default)
MapPosKeys	String	Filename of Key Map file. The default value is C:\POS\BIN\KBDKMAP.DAT.
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.
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.

POS printer

Supported settings

Table 167. Service Object settings for RS-485/USB devices (POS printer), supported

Keyword	Type	Description
PDF417ECCLevel	String	Number of error correction code words (0-400) added when printing a PDF417 bar code. The default value is 15. **Note:** This setting cannot be modified with the Configuration tool.
PDF417AspectHeight	String	Aspect ratio height (1-9) for PDF417 bar codes. The default value is 1. **Note:** This setting cannot be modified with the Configuration tool.
PDF417AspectWidth	String	Aspect ratio width (1-9) for PDF417 bar codes. The default value is 2. **Note:** This setting cannot be modified with the Configuration tool.
PDF417Truncation	String	Whether truncation is enabled for PDF417 bar codes. Valid values are: **0** Disable truncation (default) **1** Enable truncation **Note:** This setting cannot be modified with the Configuration tool.
TranslateCharacter	Key	Key values for translating one character to another. **Note:** This setting cannot be modified with the Configuration tool.
TranslateCharacter\<X>	String	One-byte decimal value of character to translate. **Note:** This setting cannot be modified with the Configuration tool.
CapRec2Color	String	Whether color thermal paper is loaded in the SureMark printer. Valid values are: **N** Color paper not loaded (default) **Y** Color paper loaded

Table 167. Service Object settings for RS-485/USB devices (POS printer), supported (continued)

Keyword	Type	Description
ProportionalFontFixedWidth	String	Alignment width, in printer dots (8-32). Used to align proportional font characters on a fixed width. The default value is 20. **Note:** This setting cannot be modified with the Configuration tool.
PrinterModel	String	4689 only. Specifies the model of 4689 printer in use. Valid values: 4689-TD5 4689-3M1 4689-3G1 4689-TD5(integrated into 4674) 4689-TG1(integrated into 4674)
DoubleWideAndDoubleHighIsQuad	String	4610-TI5 (RS-485 or USB) only. Enables conversion of double-high and double-wide escape character sequences to quad characters. Valid values are: **ON** Conversion enabled. **(other value)** Conversion disabled. Double-wide is x2-wide and x1-high. Double-high is x1-wide and x2-high.
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.
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.

Table 167. Service Object settings for RS-485/USB devices (POS printer), supported (continued)

Keyword	Type	Description
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.
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.
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.
OEMEmulation	String	Emulate OEM printer. Valid values are: **False** (default) **True**
DefaultLargeFont	String	Default to larger receipt font at startup. Valid values are: **False** (default) **True**
Default8LPI	String	Default to 8 lines per inch on receipt at startup. Valid values are: **False** (default) **True**
TraceFlags	DWORD	Used by tracing facility.
LogFile	String	
MaxSize	String	
TraceConsole	String	
DelayedWrite	String	

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 168. Service Object settings for RS-485/USB devices (POS printer), unsupported

Keyword	Type	Description
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.
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
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)
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
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)
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)

Scale

Table 169. Service Object settings for RS-485/USB devices (scale)

Keyword	Type	Description
weightMode	String	Unit of weight used by the scale. Valid values are: **0** U.S. pound (default) **1** Kilogram
RemoteDisplayAttached	String	Whether a Pole Display is attached. Valid values are: **True** Display attached **(other value)** No display (default)
ZeroScale	String	Whether zeroing the scale is supported. Valid values are: **0** Not supported (default) **1** Supported
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

Tone indicator (keyboard tone)

Table 170. Service Object settings for RS-485/USB devices (tone indicator)

Keyword	Type	Description
OnlineTimeout	String	Time (in milliseconds) to wait for device to come online. **Note:** This setting cannot be modified with the Configuration tool.

SurePOS 500/600

Cash drawer

Table 171. Service Object settings for SurePOS 500/600 devices (cash drawer)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM4. **Note:** Unit is preconfigured for COM4. Other values can be selected through the BIOS, but this is not recommended.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 9600.
CashDrawerNumber	String	Cash drawer number. Valid values are 1 (default) and 2.

Line display

Table 172. Service Object settings for SurePOS 500/600 devices (line display)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM4. **Note:** Unit is preconfigured for COM4. Other values can be selected through the BIOS, but this is not recommended.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 9600.
ControlFlow	String	Control flow for device. The only valid value is DTR/DSR.
Type	String	Line display type. Valid values are: **LINE** Single-byte display **APA** Double-byte display
Mode	String	Screen display mode. Valid values are 2x20 (default), 4x20, and 5x20.

MSR

Table 173. Service Object settings for SurePOS 500/600 devices (MSR)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM3. **Note:** Unit is preconfigured for COM3. Other values can be selected through the BIOS, but this is not recommended.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 19 200.
ControlFlow	String	Control flow for device. The only valid value is DTR/DSR.
MsrType	String	Type of MSR reader attached. Valid values are: **ISO** Standard three-track reader **JUCC** JIS-I and II reader

SurePOS 300

Cash drawer

Table 174. Service Object settings for SurePOS 300 devices (cash drawer)

Keyword	Type	Description
CashDrawerNumber	String	Cash drawer number. Valid values are 1 (default) and 2.

Line display

Table 175. Service Object settings for SurePOS 300 devices (line display)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM4.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 9600.
ControlFlow	String	Control flow for device. The only valid value is DTR/DSR.
Type	String	Line display type. Valid values are: **LINE** Single-byte display **APA** Double-byte display
Mode	String	Screen display mode. Valid values are 2x20 (default), 4x20, and 5x20.

SurePoint

MSR

Table 176. Service Object settings for SurePOS 300 devices (MSR)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM3.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 19 200.
ControlFlow	String	Control flow for device. The only valid value is DTR/DSR.
MsrType	String	Type of MSR reader attached. Valid values are: **ISO** Standard three-track reader **JUCC** JIS-I and II reader

EIA-232 attached SureMark

General settings

Table 177. Service Object settings for EIA-232 attached SureMark devices (general)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM1.
BaudRate	String	Transfer rate (in bits per second) for device. Valid values are 9600 (default) and 19 200.
ControlFlow	String	Control flow for device. Valid values are DTR/DSR (default) and XON/XOFF.

Cash drawer

Table 178. Service Object settings for EIA-232 attached SureMark devices (cash drawer)

Keyword	Type	Description
CashDrawerNumber	String	Cash drawer number. Valid values are 1 (default) and 2.
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.
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.
SignalsReversed	String	Reverses the wiring polarity of the cash drawer. Valid values are: **True** Reverse polarity **(other value)** Standard polarity (default)

MICR

Table 179. Service Object settings for EIA-232 attached SureMark devices (MICR)

Keyword	Type	Description
ExceptionFile	String	Fully-qualified filename of MICR exception file, which contains exception processing information. The default value is: C:\POS\BIN\PARSE.DAT
MinMicrSignalLevel	String	Minimum signal string (0-100) for successful MICR read. The default value is 50 (full string).

POS printer

Supported settings

Note: This setting cannot be modified with the Configuration tool.

Table 180. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported

Keyword	Type	Description
PDF417ECCLevel	String	Number of error correction code words (0-400) added when printing a PDF417 bar code. The default value is 15. **Note:** This setting cannot be modified with the Configuration tool.
PDF417AspectHeight	String	Aspect ratio height (1-9) for PDF417 bar codes. The default value is 1. **Note:** This setting cannot be modified with the Configuration tool.

Table 180. Service Object settings for EIA-232 attached SureMark devices (POS printer), supported (continued)

Keyword	Type	Description
PDF417AspectWidth	String	Aspect ratio width (1-9) for PDF417 bar codes. The default value is 2. **Note:** This setting cannot be modified with the Configuration tool.
PDF417Truncation	String	Whether truncation is enabled for PDF417 bar codes. Valid values are: **0** Disable truncation (default) **1** Enable truncation **Note:** This setting cannot be modified with the Configuration tool.
TranslateCharacter	Key	Key values for translating one character to another. **Note:** This setting cannot be modified with the Configuration tool.
TranslateCharacter\<X>	String	One-byte decimal value of character to translate. **Note:** This setting cannot be modified with the Configuration tool.
CapRec2Color	String	Whether color thermal paper is loaded in the SureMark printer. Valid values are: **N** Color paper not loaded (default) **Y** Color paper loaded
ProportionalFontFixedWidth	String	Alignment width, in printer dots (8-32). Used to align proportional font characters on a fixed width. The default value is 20.
OEMEmulation	String	Emulate OEM printer. Valid values are: **False** (default) **True**
DefaultLargeFont	String	Default to larger receipt font at startup. Valid values are: **False** (default) **True**
Default8LPI	String	Default to 8 lines per inch on receipt at startup. Valid values are: **False** (default) **True**

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 181. Service Object settings for EIA-232 attached SureMark devices (POS printer), unsupported

Keyword	Type	Description
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.
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
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

SureOne devices

Line display

Table 182. Service Object settings for SureOne devices (line display)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM2. **Note:** Unit is preconfigured for COM2. Other values can be selected through the BIOS, but this is not recommended.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 9600.
ControlFlow	String	Control flow for device. The only valid value is DTR/DSR.

POS printer

Table 183. Service Object settings for SureOne devices (POS printer)

Keyword	Type	Description
ComPort	String	Com port for device. Valid values are COM1 through COM6. The default value is COM1. **Note:** Unit is preconfigured for COM1. Other values can be selected through the BIOS, but this is not recommended.
BaudRate	String	Transfer rate (in bits per second) for device. The only valid value is 9600.
ControlFlow	String	Control flow for device. Valid values are DTR/DSR (default) and XON/XOFF.

Table 183. Service Object settings for SureOne devices (POS printer) (continued)

Keyword	Type	Description
ThermalPrinter	String	Whether printer is impact or thermal. Valid values are: **0** Impact (default) **1** Thermal
NarrowPaper	String	Whether paper is normal or narrow width. Valid values are: **0** Normal (default) **1** Narrow
BitmapResolution	String	Bitmap resolution (for thermal printers only). Valid values are: **0** Low resolution (default) **9** High resolution
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

Part 3. Keyboards and code pages

Chapter 8. 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 (in JavaPOS terms, a KEY_DOWN event), the keyboard device driver receives a code that is called a *make scan code*. There are different codes for each key. When a key is released on some keyboards (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 using the code page that the application is using.

Checkout keyboards (50-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

50-Key Modifiable Layout Keyboard

Figure 32 shows the key-switch numbers. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

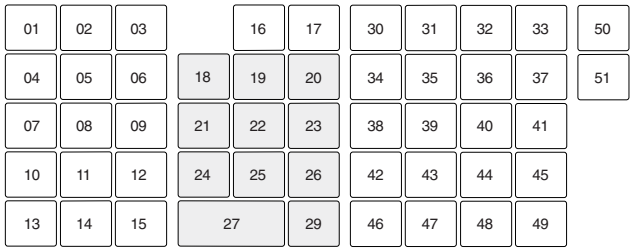


Figure 32. 50-Key Modifiable Layout Keyboard

50-Key Modifiable Layout Keyboard and Operator Display

Figure 33 shows the key-switch numbers. The numeric keypad (key switches 18 to 29) is shown in the shaded area of the illustration.

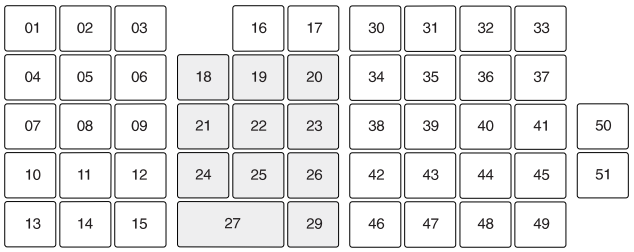


Figure 33. 50-Key Modifiable Layout Keyboard and Operator Display

Retail Point of Sale Keyboard Layout

Figure 34 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.

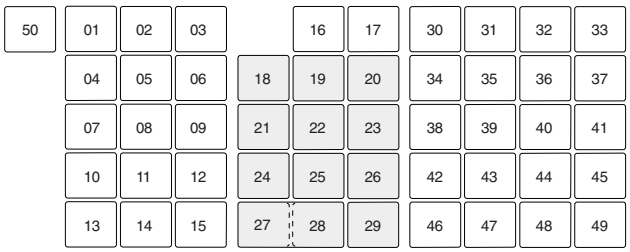


Figure 34. Retail Point of Sale Keyboard

Retail Point of Sale Keyboard with Card Reader and Display

Figure 35 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.

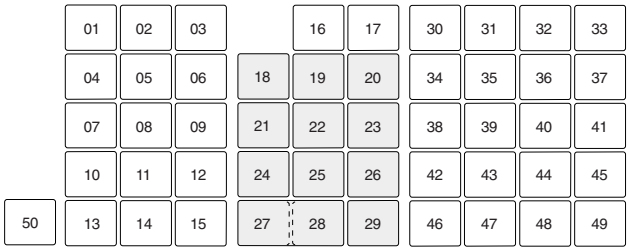


Figure 35. Retail Point of Sale Keyboard With Card Reader and Display

Point of Sale Keyboard VI Layout

Figure 36 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.

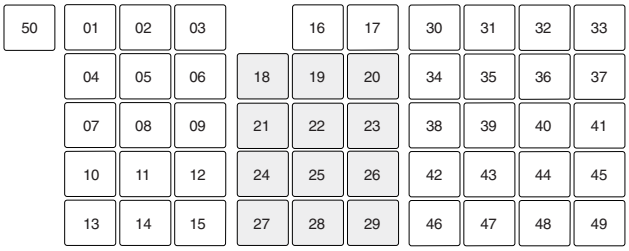


Figure 36. Point of Sale Keyboard VI

Checkout Keyboards RS-485/USB Scan Code Set

Table 184 on page 242 shows the key scan codes for the following keyboards:

- 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

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 184. 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'	
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'	

Table 184. Checkout Keyboards RS-485/USB Scan Code Set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	
	+ 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

Table 185 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 185. 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 37 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.

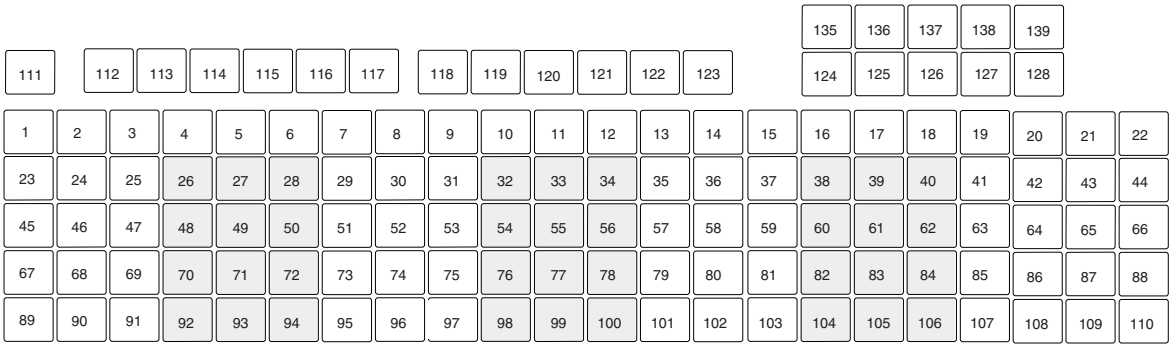


Figure 37. Modifiable Layout Keyboard with Card Reader

Modifiable Layout Keyboard with Card Reader RS-485/USB Scan Code Set

Table 186 shows the key scan codes for the Modifiable Layout Keyboard with Card Reader.

Table 186. 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 186. 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 186. 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 186. 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 186. 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 187 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 187. 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)

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)

PC Point of Sale keyboard with Card Reader (ANKPOS)

Figure 38 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.

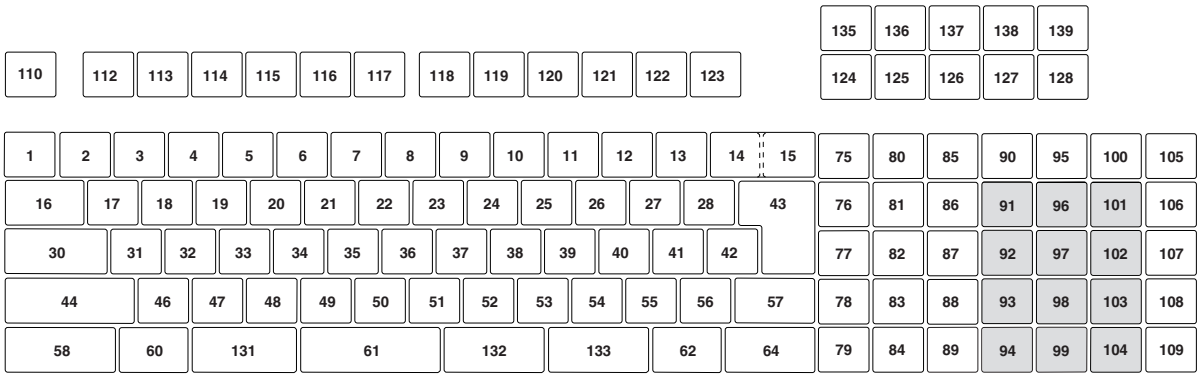


Figure 38. PC Point of Sale Keyboard Layout

Retail Alphanumeric Point of Sale Keyboard with Card Reader (NANPOS)

Figure 39 shows the layout and assigned key-switch numbers for the Retail Alphanumeric Point of Sale Keyboard with Card Reader.

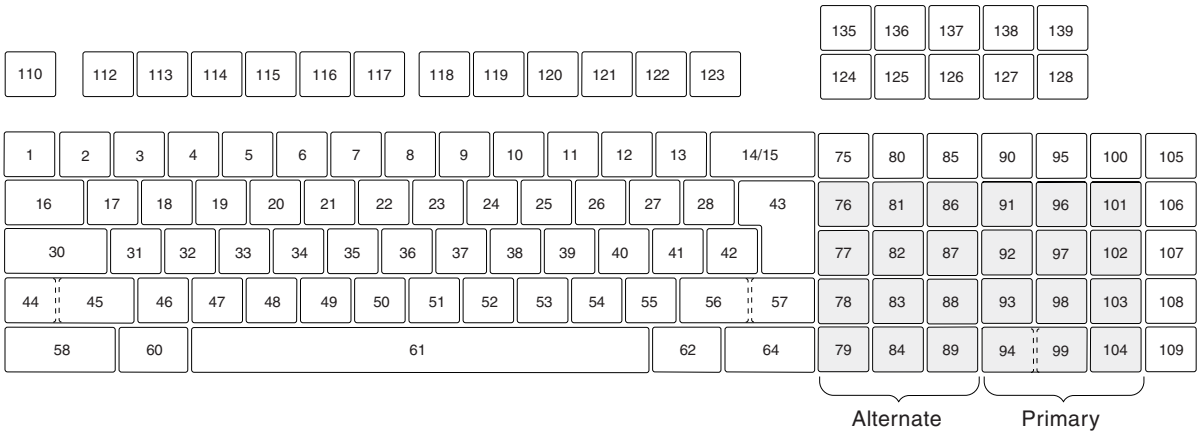


Figure 39. Layout and Assigned Switch Numbers

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 188 on page 251 for the scan codes associated with these keys.

RS-485/USB Scan Code Set for the Retail Alphanumeric Point of Sale Keyboard (NANPOS) and the PC Point of Sale Keyboard (ANKPOS)

Table 188 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the RS-485 or USB port.

Table 188. 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'	
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'	

Table 188. RS-485/USB Scan Code Set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	

Table 188. RS-485/USB Scan Code Set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	

Table 188. RS-485/USB Scan Code Set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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

Table 188. RS-485/USB Scan Code Set (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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 189 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 189. 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 190 shows the scan codes sent by the S1 and S2 function keys, depending upon the Keylock Positions.

Table 190. 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'

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)

Retail Alphanumeric Point of Sale with Card Reader or PC Point of Sale keyboard (ANKPOS) Layout Keyboard

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.

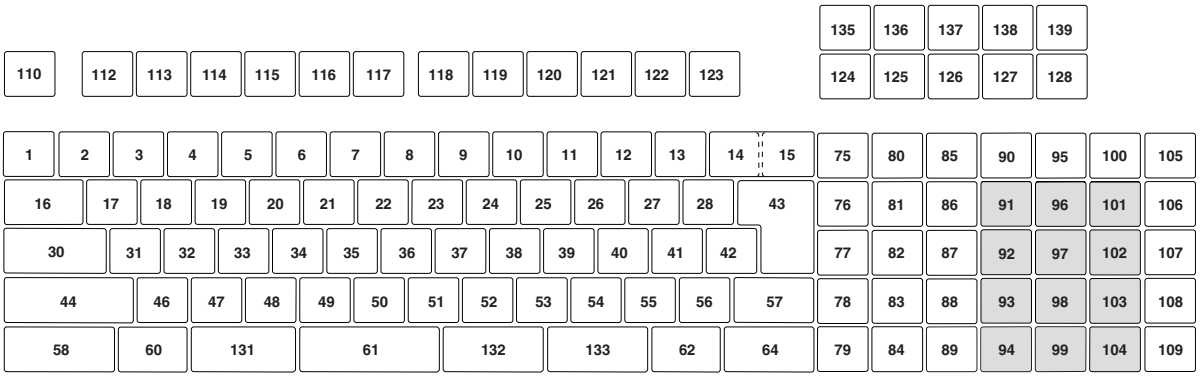


Figure 40. PC Point of Sale Keyboard Layout

Retail Alphanumeric Point of Sale with Card Reader (NANPOS) Layout Keyboard

Figure 41 shows the layout and assigned key-switch numbers for the Retail Alphanumeric Point of Sale Keyboard with Card Reader.

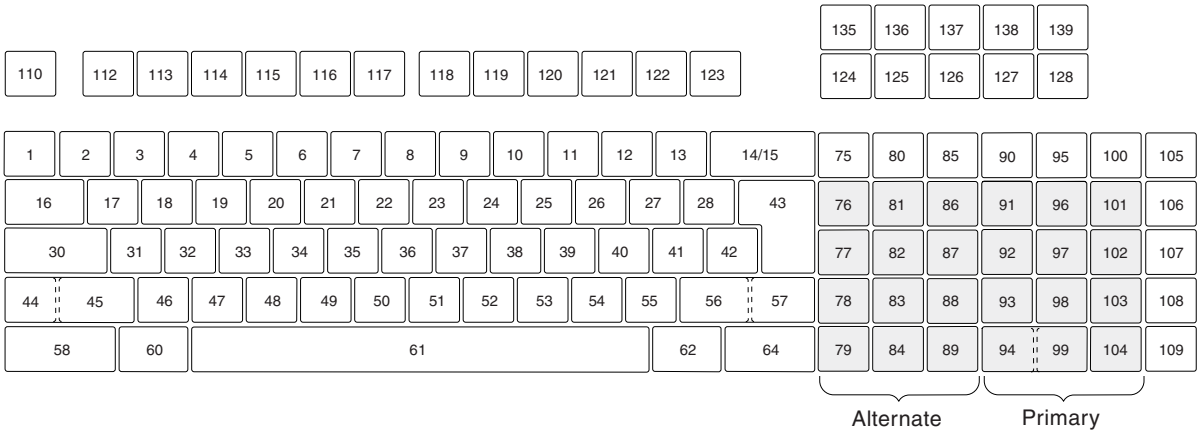


Figure 41. Layout and Assigned Switch Numbers

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 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 191 on page 259 for the scan codes associated with these keys.

Compact Alphanumeric Point of Sale (CANPOS) Layout Keyboard

Figure 42 shows the layout and assigned key-switch numbers for the Compact Alphanumeric Point of Sale Keyboard with Card Reader.

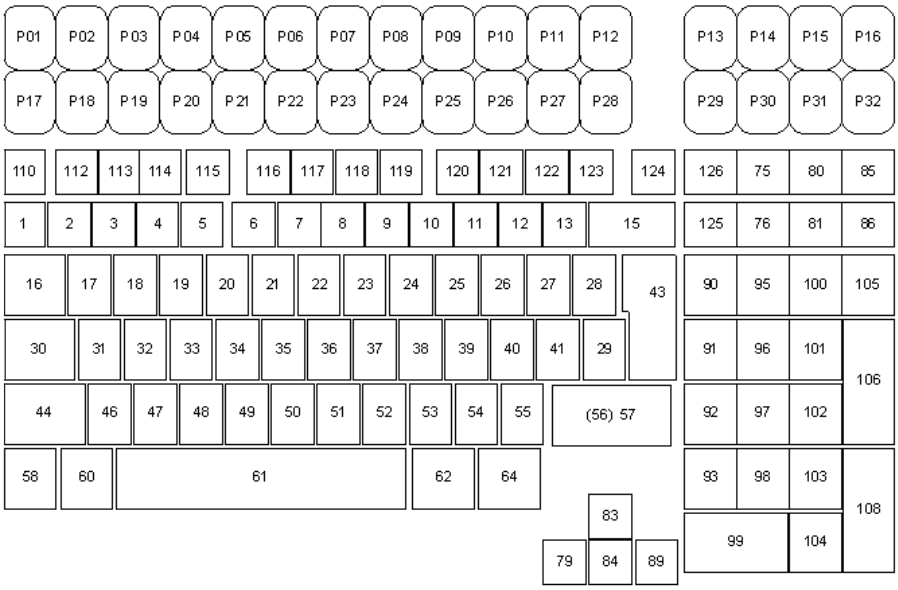


Figure 42. Layout and Assigned Switch Numbers (CANPOS)

Note: See Table 192 on page 266 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) and the PC Point of Sale Keyboard (ANKPOS)

Table 191 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 191. 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'	
35	X'22'	X'a2'	X'22'	

Table 191. PS/2 or USB (System Attached) Scan Codes (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'	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'	
86	X'e0' X'51'	X'e0' X'd1'	X'01' X'51'	

Table 191. PS/2 or USB (System Attached) Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	
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'	

Table 191. PS/2 or USB (System Attached) Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'37'	X'b7'	X'37'	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'01' X'1c'	Key press only
135	X'9d' X'01' X'1d'		X'01' X'01'	Key press only
	+ Right Ctrl Case Pressed			
124	X'e0' X'9d' X'1c' X'e0' X'1d'		X'01' X'1c'	Key press only
135	X'e0' X'9d' X'01' X'e0' X'1d'		X'01'	Key press only
	+ Both Ctrl Case Pressed			
124	X'e0' X'9d' X'9d' X'1c' X'e0' X'1d'		X'01' X'1c'	Key press only
135	X'e0' X'9d' X'9d' X'01' X'e0' X'1d'		X'01'	Key press only
	+ 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			

Table 191. PS/2 or USB (System Attached) Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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			

Table 191. PS/2 or USB (System Attached) Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	

Table 191. PS/2 or USB (System Attached) Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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)

Table 192 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 192. PS/2 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'	
36	X'23'	X'a3'	X'23'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
92	X'4b'	X'cb'	X'4b'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
93	X'4f'	X'cf'	X'4f'	
95	X'e0' X'35'	X'e0' X'b5'	X'01' X'35'	
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'	
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'4a'	X'ca'	X'4a'	
106	X'4e'	X'ce'	X'4e'	
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'	
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
	+ Left Alt Case Pressed			
124	X'37'	X'b7'	X'37'	
	+ Right Alt Case Pressed			
124	X'37'	X'b7'	X'37'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
	+ Both Alt Case Pressed			
124	X'37'	X'b7'	X'37'	
	+ Left or Right Ctrl Pressed			
126	X'e0' X'46' X'e0' X'c6'		none	
	+ Left Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	
	+ Right Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	
	+ Both Ctrl or Shift Case Pressed			
124	X'e0' X'37'	X'e0' X'b7'	X'01' X'37'	
	+ 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'	
	+ 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'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	
	+ 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'	
	+ 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'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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'	
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'	

Table 192. PS/2 Scan Codes for CANPOS Keyboard (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
P14	X'e0' X'20'	X'e0' X'a0'	X'01' X'20'	
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'	

Point of Sale Keyboard V

Figure 43 shows the Keyboard V layout.

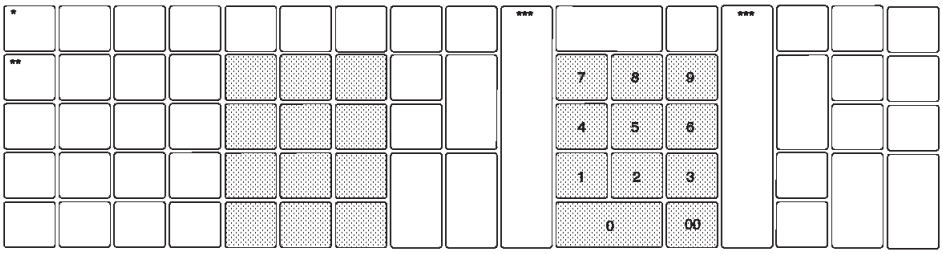


Figure 43. Point of Sale Keyboard V Layout

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 44 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 44. 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 45 shows the PLU Keyboard and Display-III layout and scan codes.

The two possible locations for the numeric keypad are shaded in Figure 45. 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 45. 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 46 shows the layout of the keyboard.

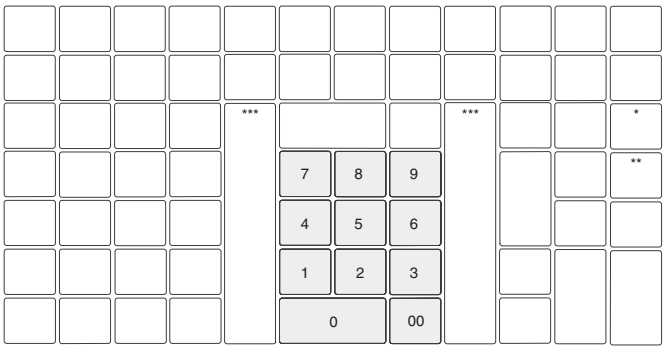


Figure 46. 4674 Point of Sale Keyboard (Built-In)

Note: The triple asterisk (***) indicates that a 1×5 dummy cap covers those key switches.

4674 POS Keyboard (Built-In) Scan Codes

Figure 47 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 47. 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 48 shows the layout of the 4685 Keyboard Model K01.

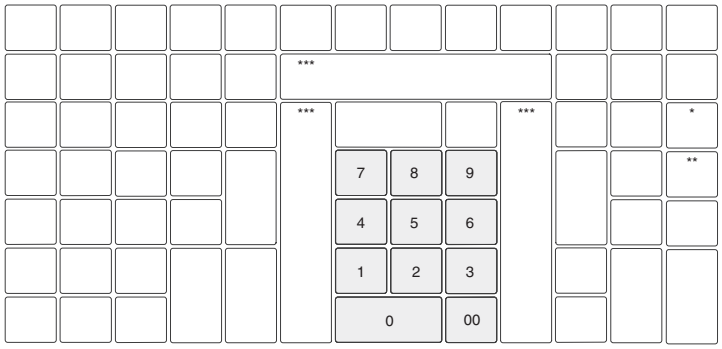


Figure 48. 4685 Point of Sale Keyboard Model K01 Layout

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 49 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 49. 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 (the key scan code without parentheses in the illustration), unlike the single-byte character set keyboards.

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 50 shows the layout of the keyboard.

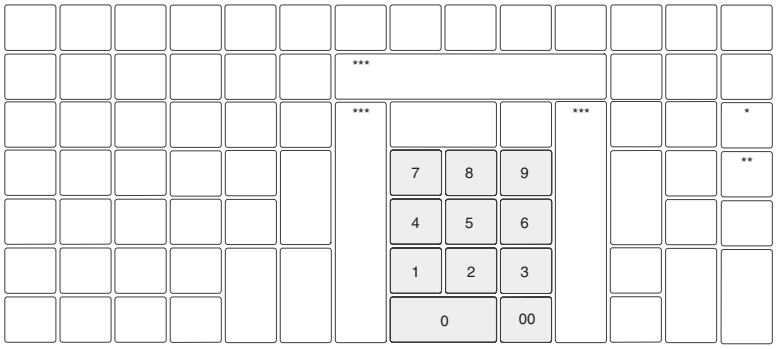


Figure 50. 4685 Keyboard K02 Ultra 7 with 4-position keylock

4685 Keyboard K02 Ultra 7 Scan Codes

Figure 51 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 51. 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 52 shows the layout of the keyboard.

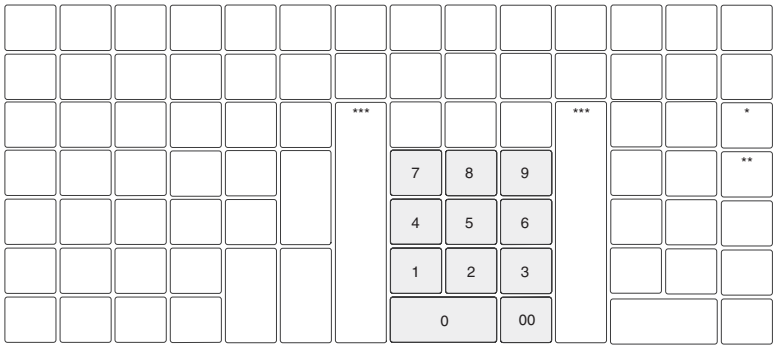


Figure 52. 4685 Keyboard K02 Ultra 7 with MSR/E (4 or 6 position keylock)

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 53 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 53. 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 54 shows the keyboard layout.

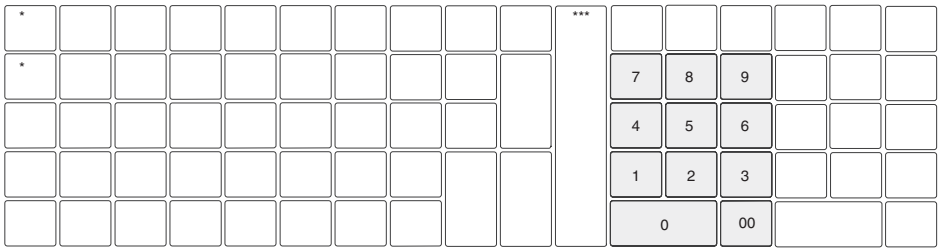


Figure 54. 4685 Model KC1

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 55 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 55. 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 56 shows the layout of the keyboard.

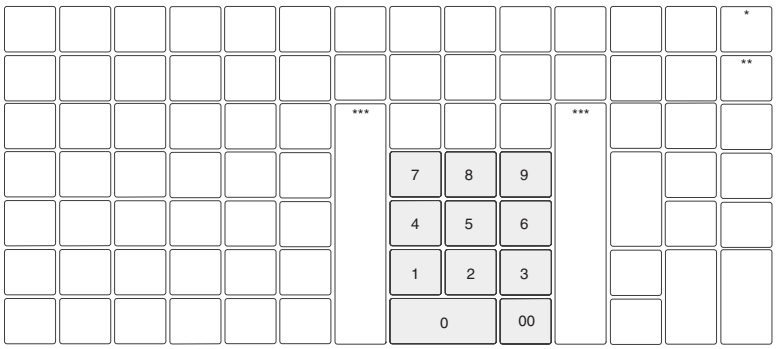


Figure 56. 4685 Keyboard Model K03 layout

Notes:

1. A single asterisk (*) indicates the S1 key.
2. Double asterisks (**) indicate the S2 key.
3. Triple asterisks (***) indicate the S3 key.

4685 Keyboard Model K03 scan codes

Figure 57 provides the scan codes.

0x6C	0x6B	0x6A	0x69	0x68	0x67	0x66	0x65	0x64	0x63	0x62	0x6F	0x6E	0x6D (0x00)
0x5C	0x5B	0x5A	0x59	0x58	0x57	0x56	0x55	0x54	0x53	0x52	0x5F	0x5E	0x5D (0x01)
0x4C	0x4B	0x4A	0x49	0x48	0x47	<0x46>	0x45	<0x44>	0x43	<0x42>	0x4F	0x4E	0x4D
0x3C	0x3B	0x3A	0x39	0x38	0x37	<0x36>	0x35	0x34	0x33	<0x32>	<0x3F>	0x3E	0x3D
0x2C	0x2B	0x2A	0x29	0x28	0x27	<0x26>	0x25	0x24	0x23	<0x22>	0x2F	0x2E	0x2D
0x1C	0x1B	0x1A	0x19	0x18	0x17	<0x16>	0x15	0x14	0x13	<0x12>	<0x1F>	<0x1E>	<0x1D>
0x0C	0x0B	0x0A	0x09	0x08	0x07	<0x06>	0x05	0x04	0x03	<0x02>	0x0F	0x0E	0x0D

Figure 57. 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.

4820 IBM SurePoint Solution Keypad

Figure 58 shows the layout of the key-switch numbers for the 4820 Keypad.

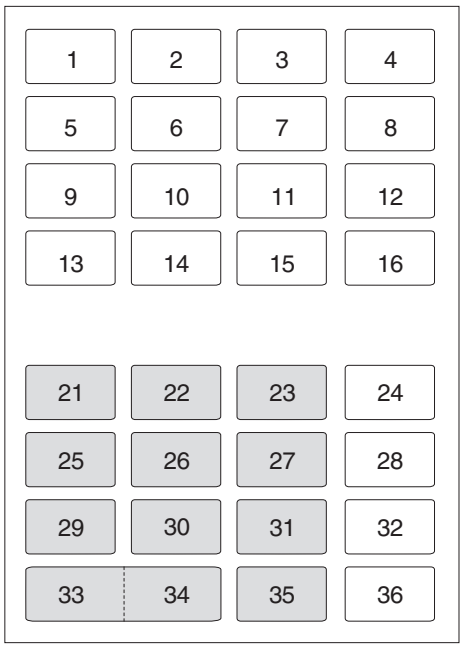


Figure 58. 4820 Keypad Key-Switch Layout

Table 193 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the RS-485 or USB port.

Table 193. 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'	
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'	
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'	
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'	

Table 193. 4820 RS-485/USB IBM Surepoint Solution Keypad Scan Codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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 59 shows the layout of the key-switch numbers for the 4820 Keypad.

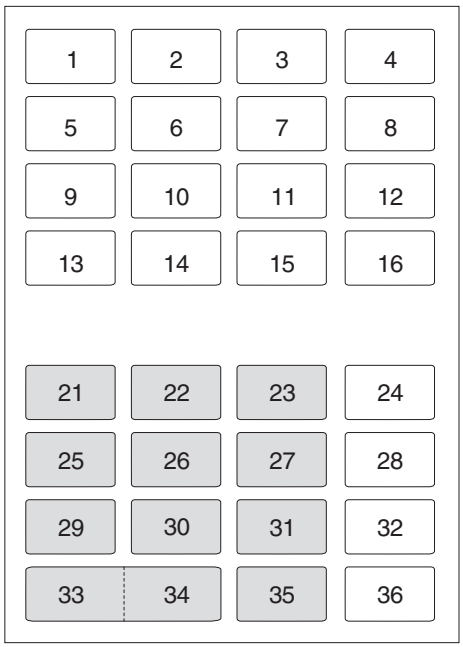


Figure 59. 4820 Keypad (system-attached) key-switch layout

Table 194 relates the keyboard key-switch number to the scan codes received when the keyboard is attached to the system keyboard port.

Table 194. 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 keyboard (built-in)

Figure 60 shows the layout of the key-switch numbers for the SureOne keyboard.

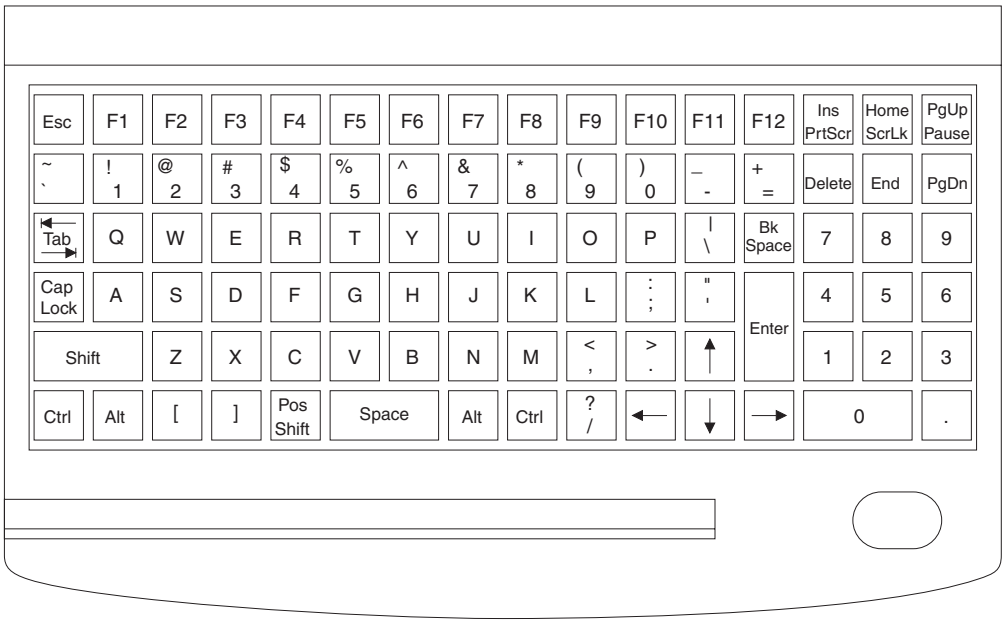


Figure 60. SureOne keyboard (built-in) key-switch layout

SureOne keyboard (built-in) scan codes

Figure 61 provides the key switch numbers.

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 61. SureOne keyboard (built-in) key switch numbers

Table 195 relates the keyboard key switch number to the scan codes received from the keyboard.

Table 195. 4820 IBM SurePoint Solution Keypad (system-attached) 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`	

Table 195. 4820 IBM SurePoint Solution Keypad (system-attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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`	
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`	

Table 195. 4820 IBM SurePoint Solution Keypad (system-attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
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`	
96	0x48`	0xc8`	0x48`	
97	0x4c`	0xcc`	0x4c`	
98	0x50`	0xd0`	0x50`	
99	0x77`	0xf7`	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`	

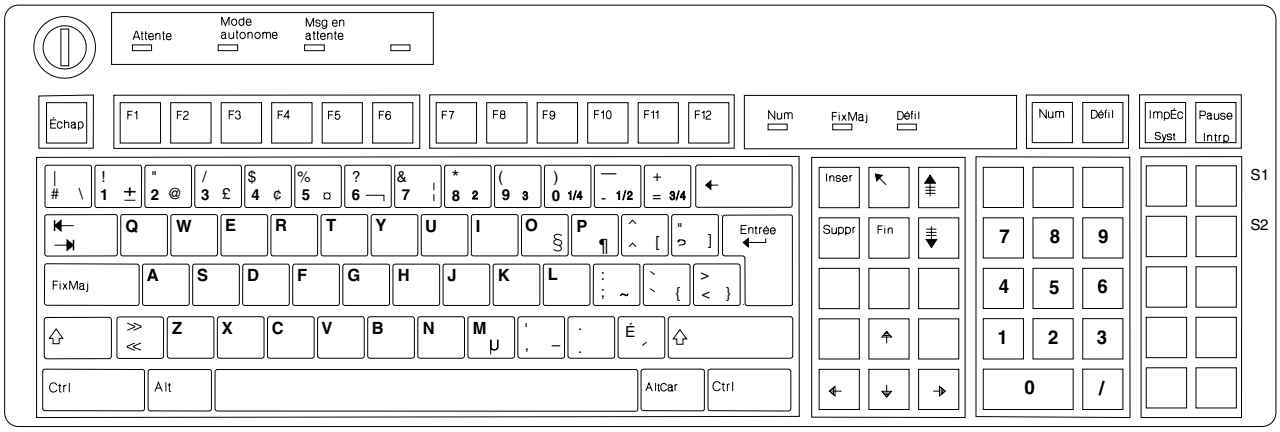
Table 195. 4820 IBM SurePoint Solution Keypad (system-attached) scan codes (continued)

Key switch number	Hardware make code	Hardware break code	JavaPOS - POSKeyData	Comments
	+ Ins = (136-PrtScr)			
(*1) Pos Shift		0xe0 0xb7 0xe0 0xaa`	0x0137`	Key release only
	+ Home = (137-ScrLk)			
(*1)	0x46`	0xc6`	0x46`	
	+ PgUp = (138-pause)			
(*1)	0xe1 0x1d 0x45 0xe1 0x9d 0xc5`		0x45`	Key press only

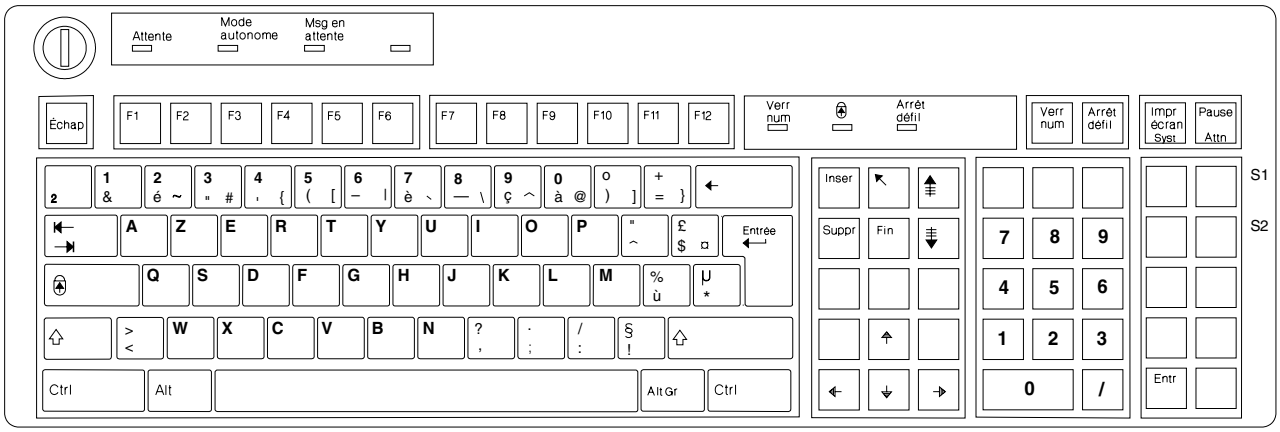
Alphanumeric Point-of-Sale (NANPOS) Country-Dependent Keyboards

The following pages illustrate the keyboards for all the supported NANPOS keyboards.

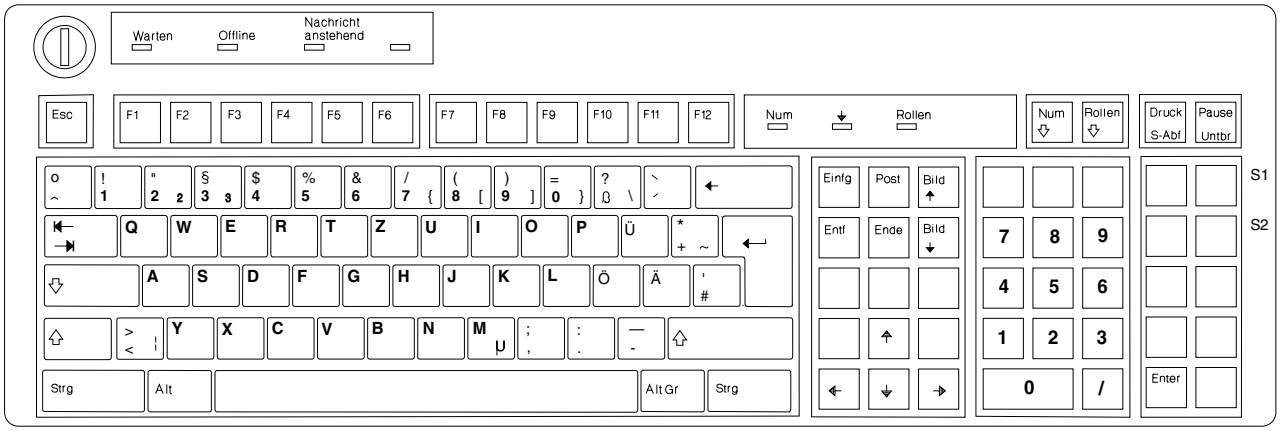
Canadian French Keyboard Layout



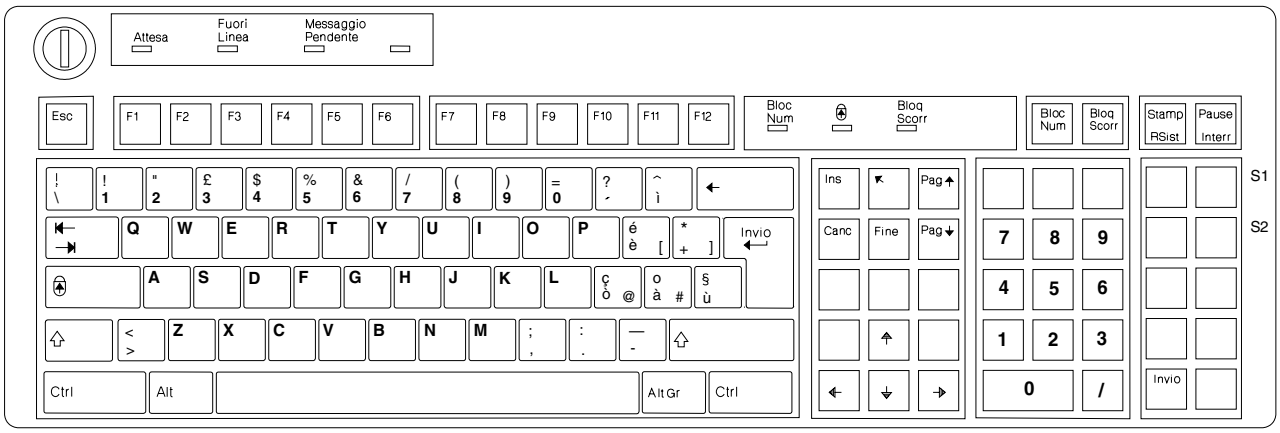
French Keyboard Layout



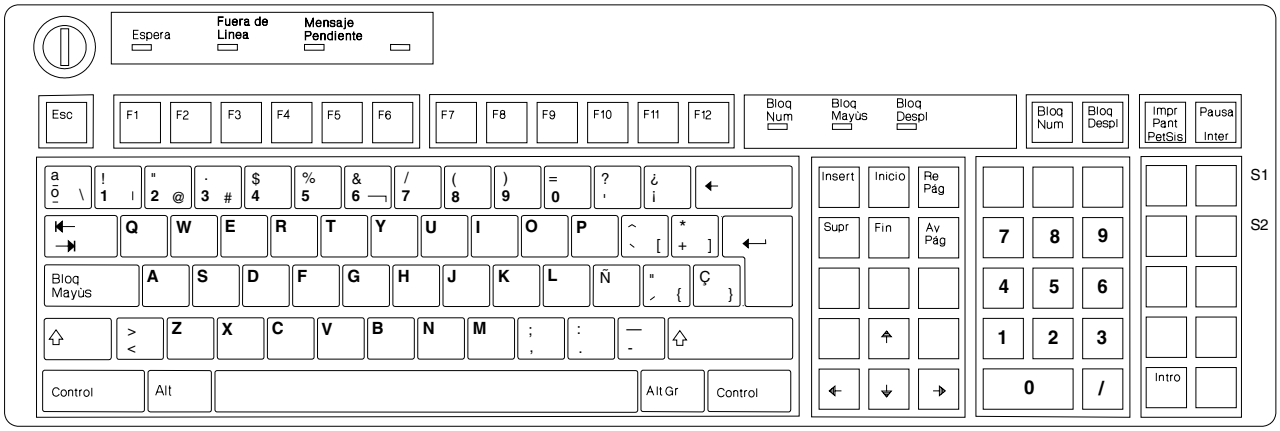
German Keyboard Layout



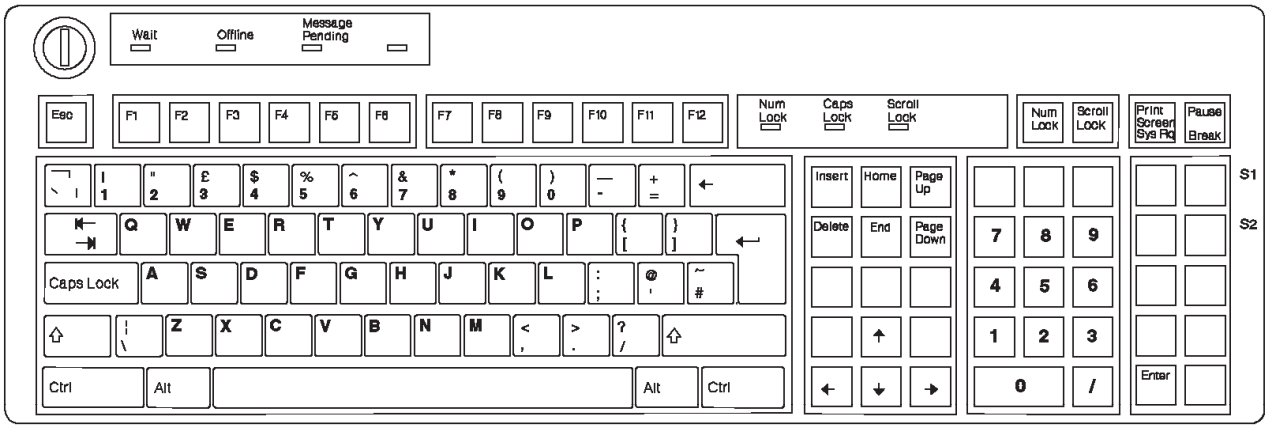
Italian Keyboard Layout



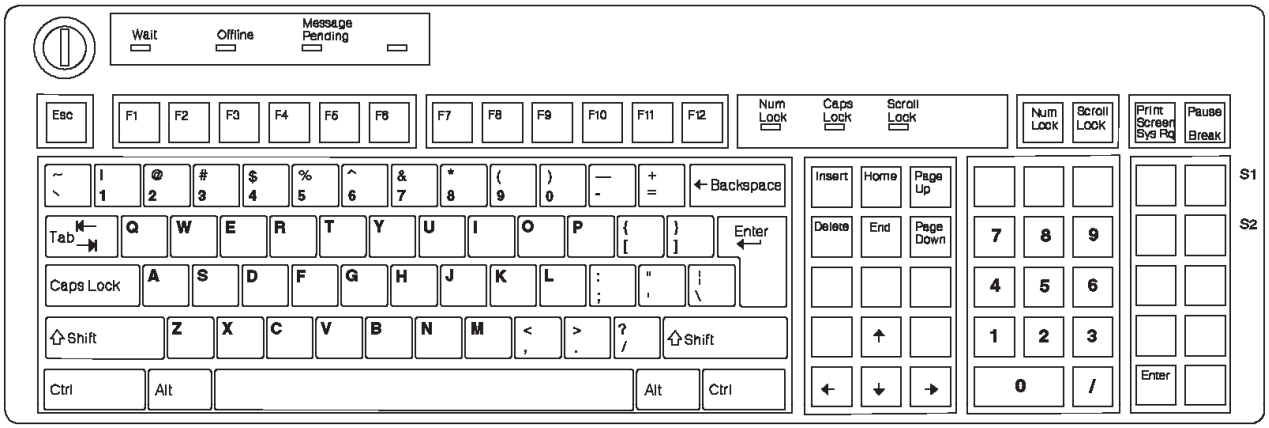
Spanish Keyboard Layout



U.K. English Keyboard Layout



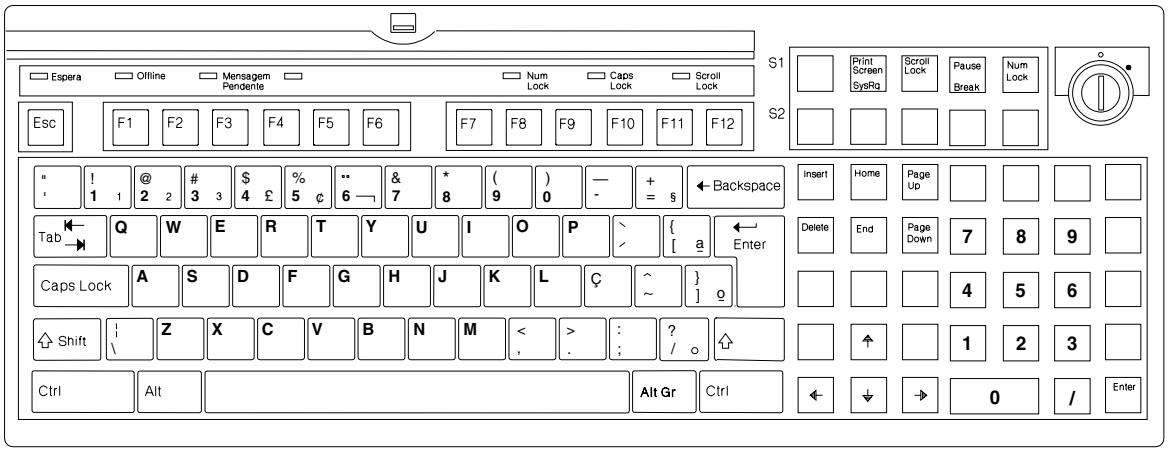
U.S. English Keyboard Layout



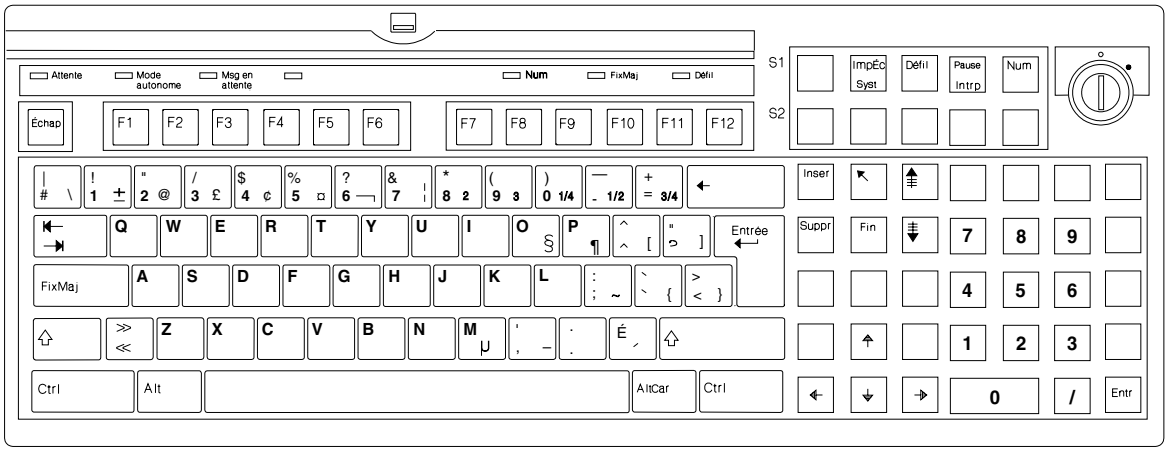
Retail Alphanumeric Point of Sale Country Dependent Keyboards

The following pages illustrate the keyboards for all the supported Retail Alphanumeric Point of Sale Keyboards.

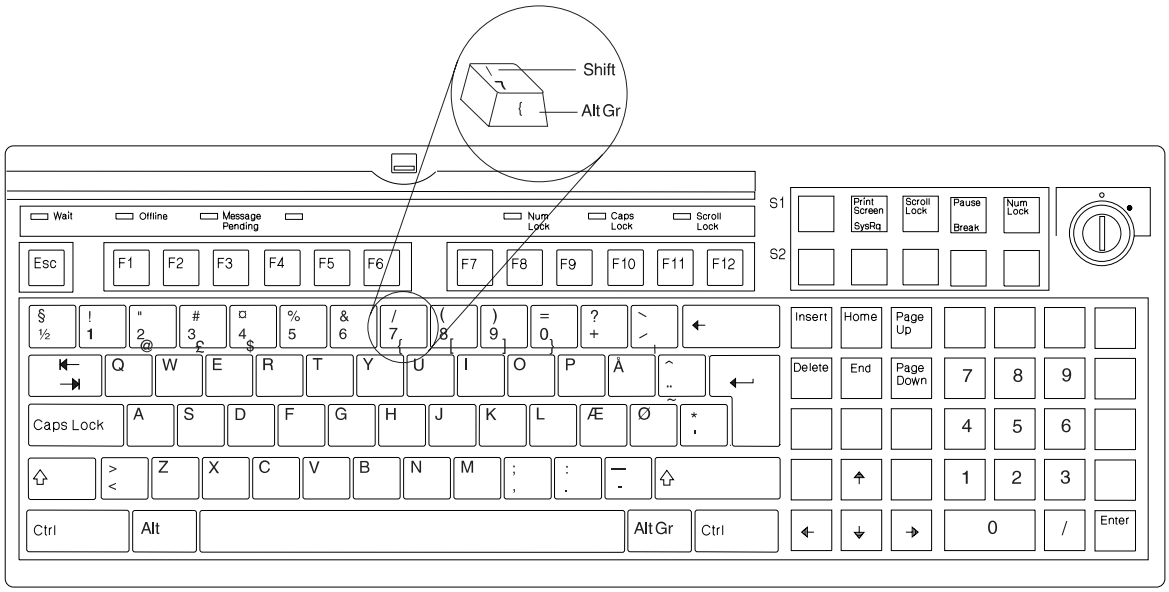
Brazil-Portuguese Keyboard Layout



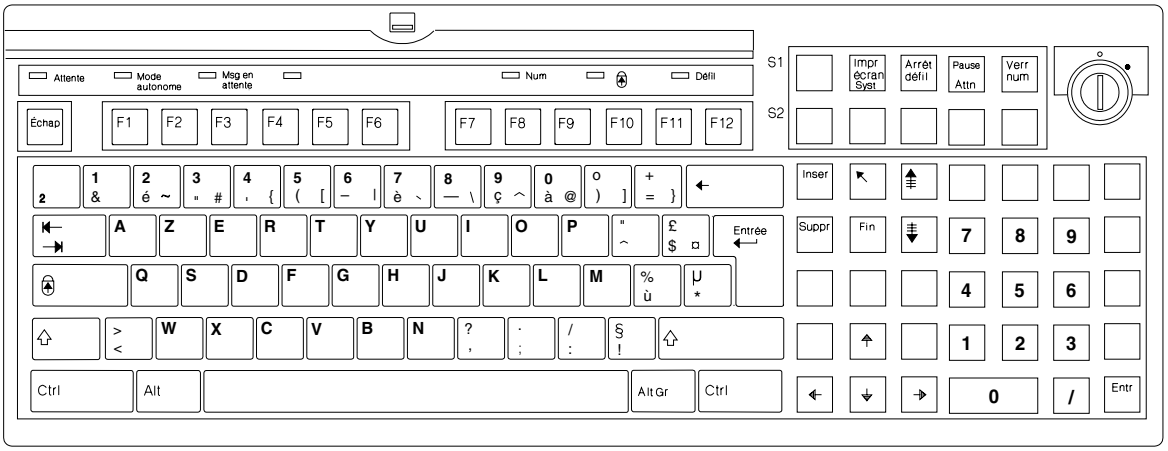
Canadian French Keyboard Layout



Danish Keyboard Layout

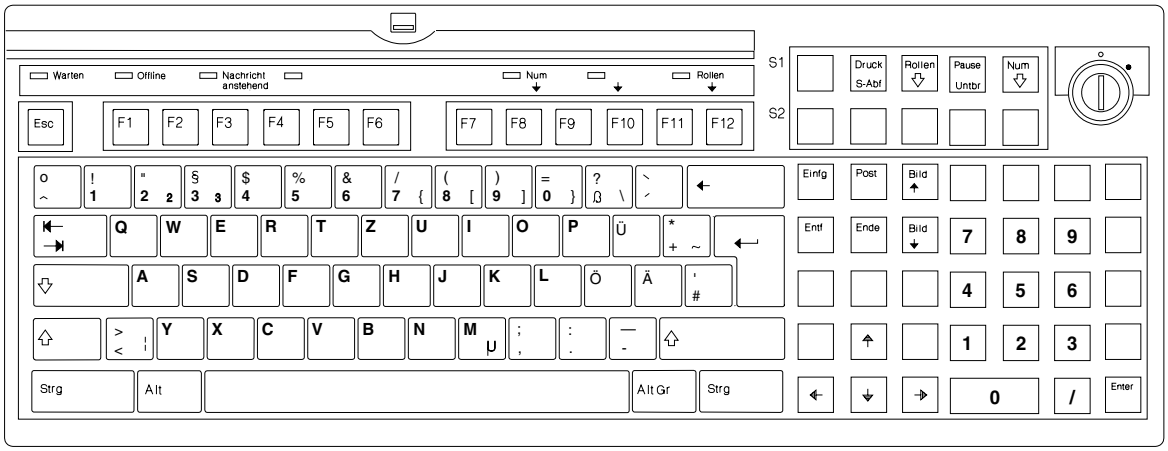


French Keyboard Layout

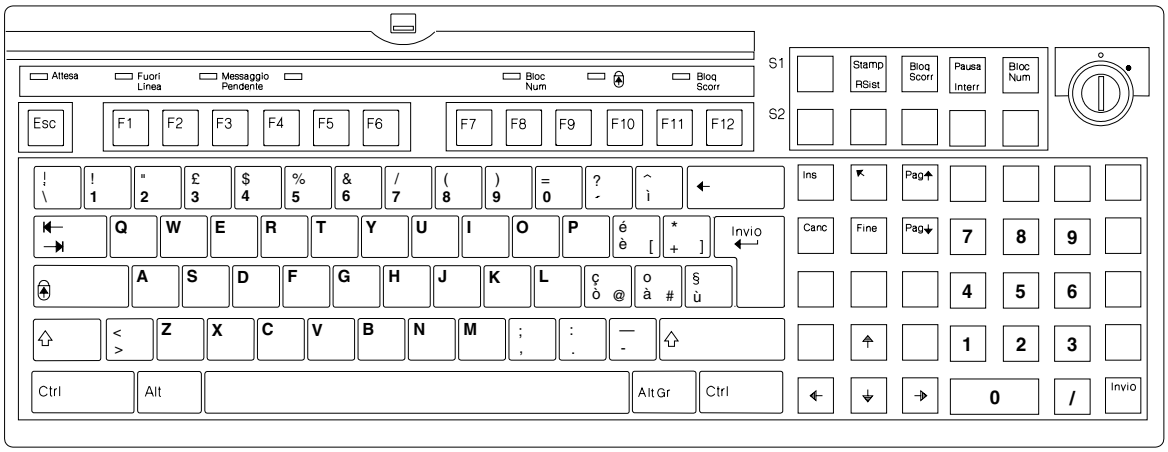


last update: August 29, 2006

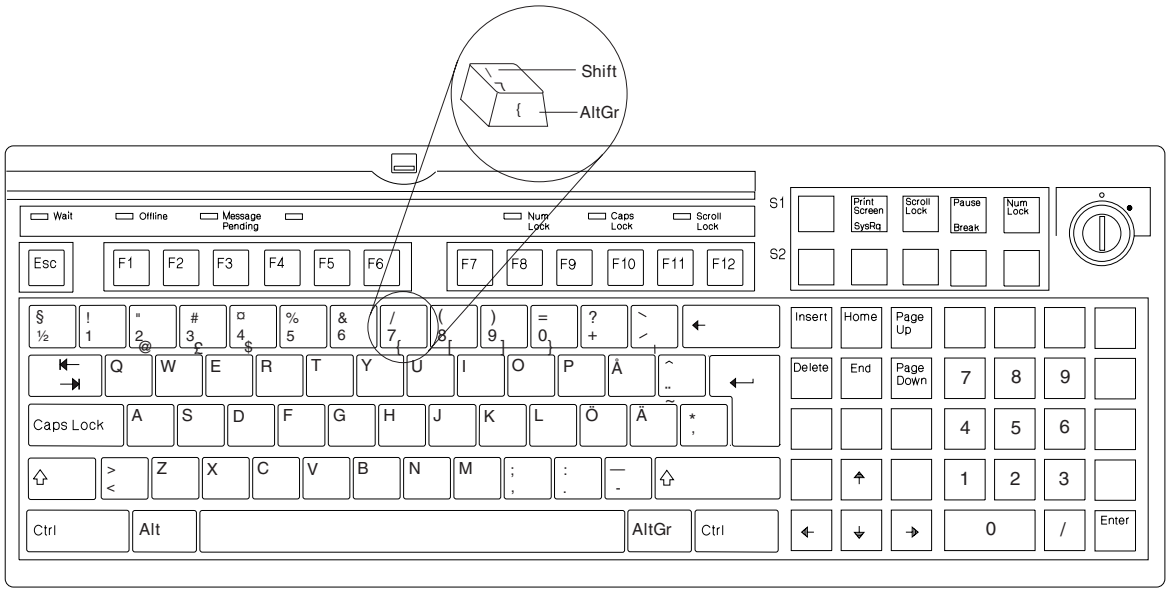
German Keyboard Layout



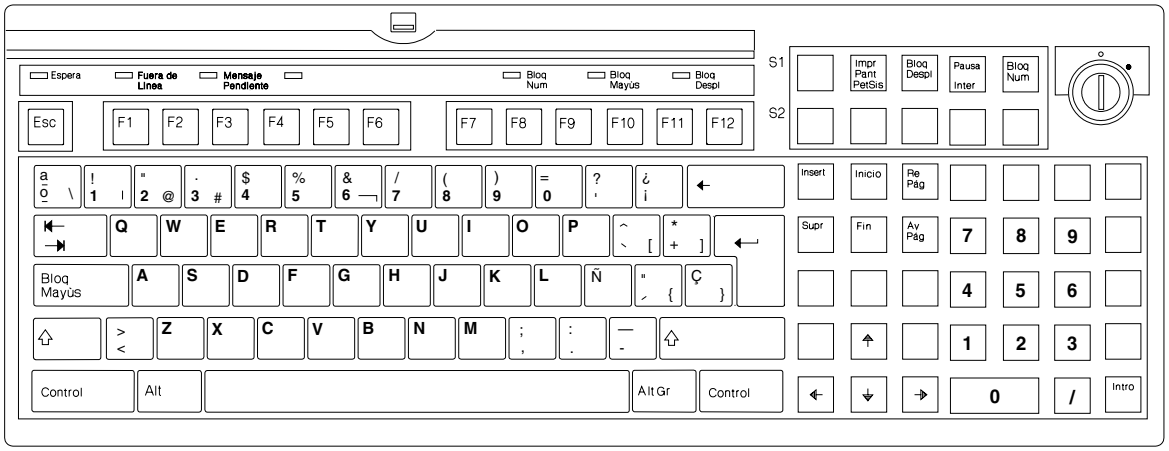
Italian Keyboard Layout



Norwegian Keyboard Layout

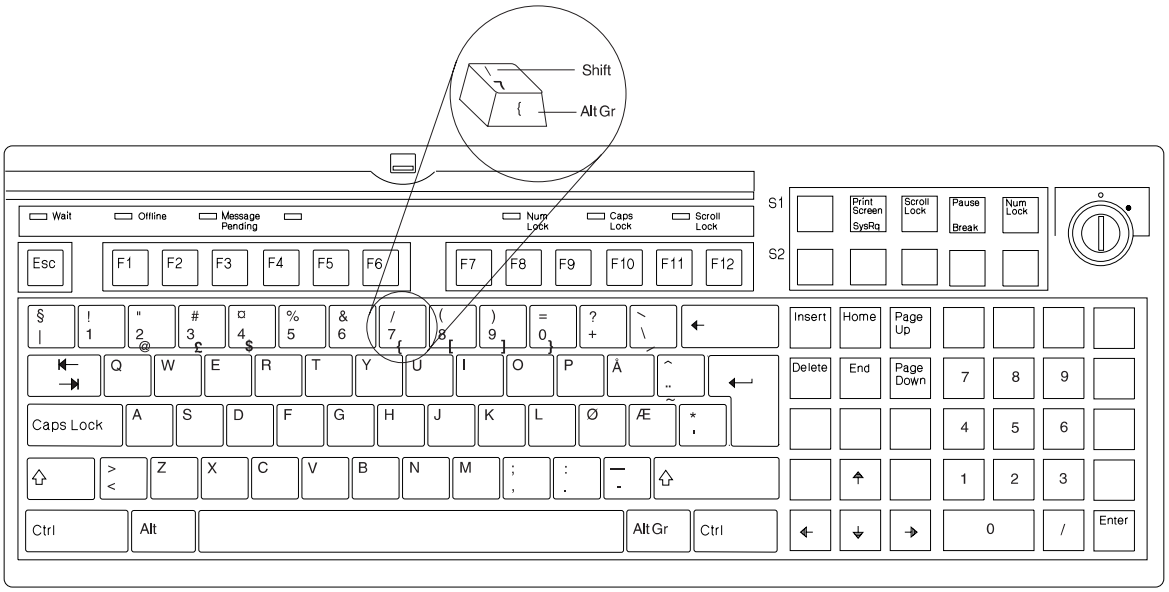


Spanish Keyboard Layout

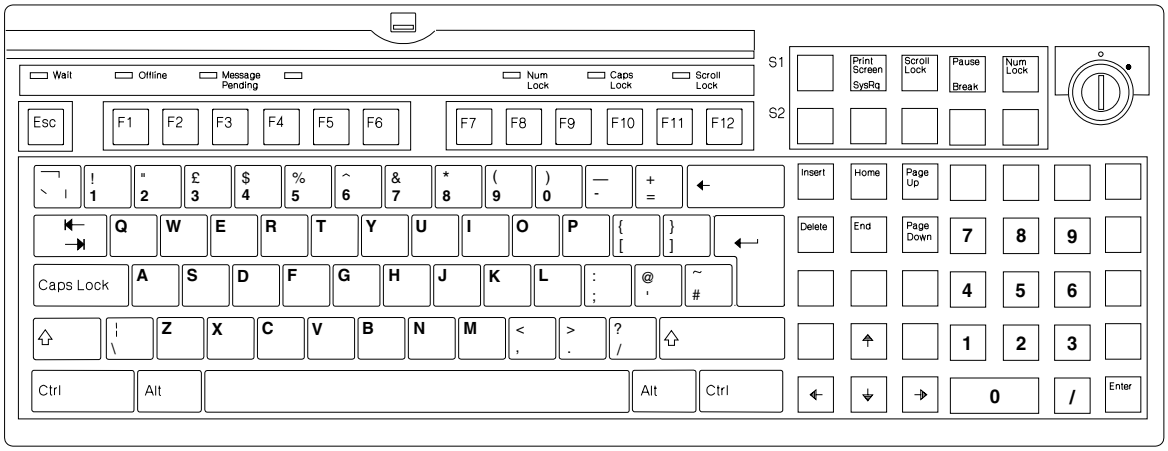


last update: August 29, 2006

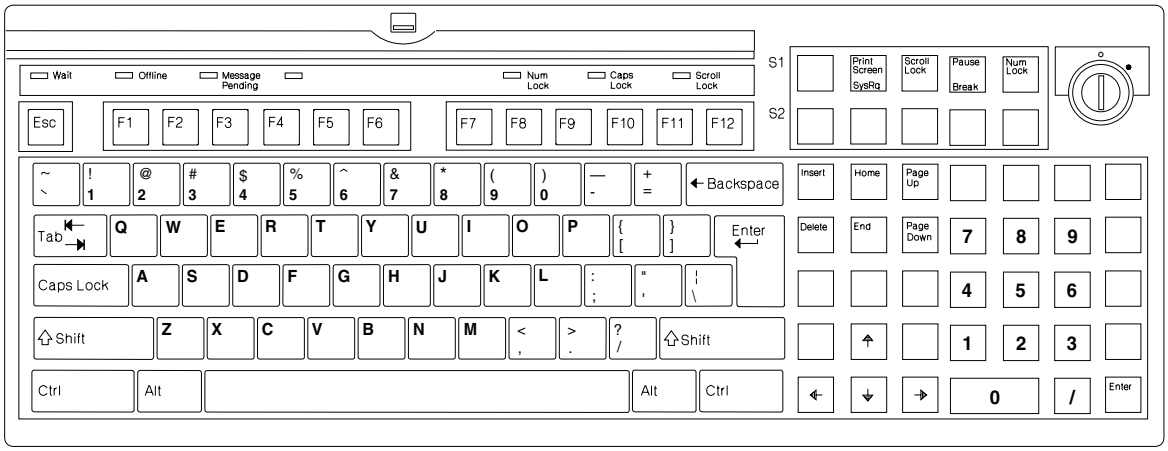
Swedish and Finnish Keyboard Layout



U.K. English Keyboard Layout



U.S. English Keyboard Layout



Alphanumeric Point-of-Sale (CANPOS) Keyboard

This section describes the layout and assigned key-switch numbers for the NANPOS keyboards

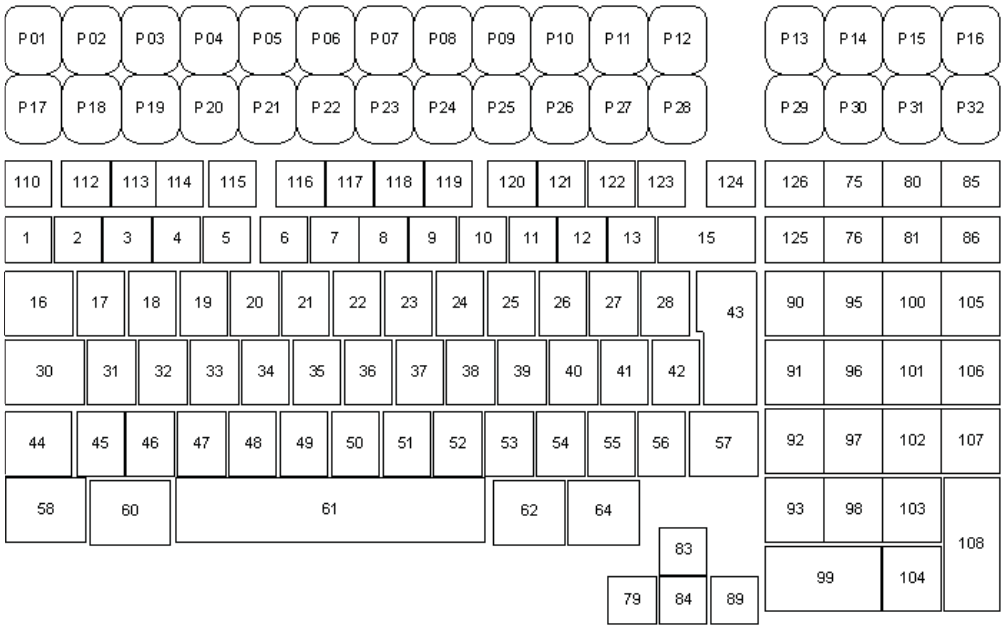
Compact Alphanumeric Point-of-Sale (CANPOS) Keyboards

This section describes the keyboard layout for the Compact Alphanumeric Point of Sale (CANPOS) keyboard.

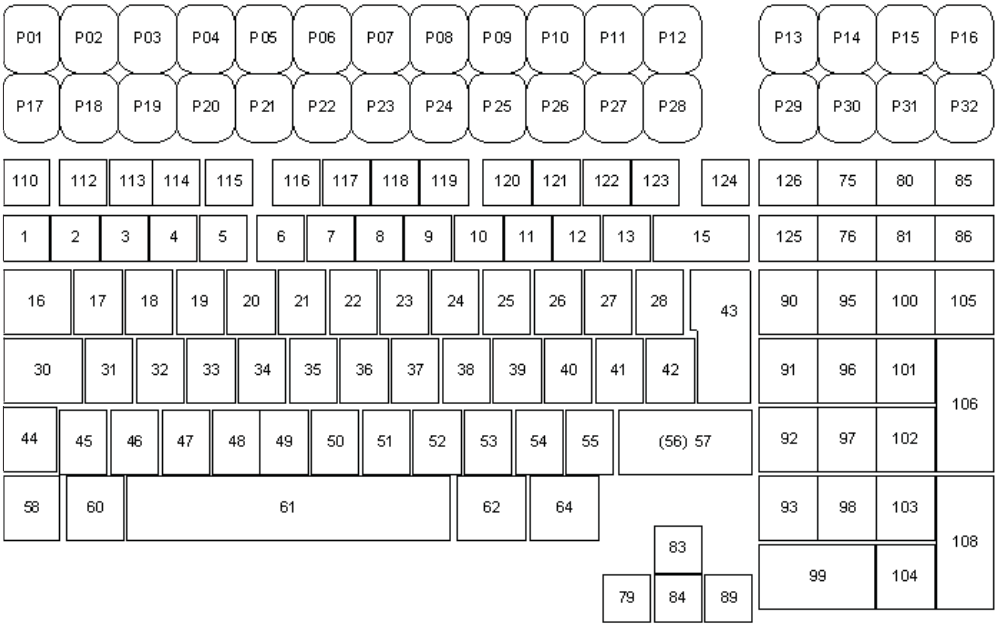
Scan Code Keyboard Assignments

This section shows the CANPOS scan code keyboard assignment by country.

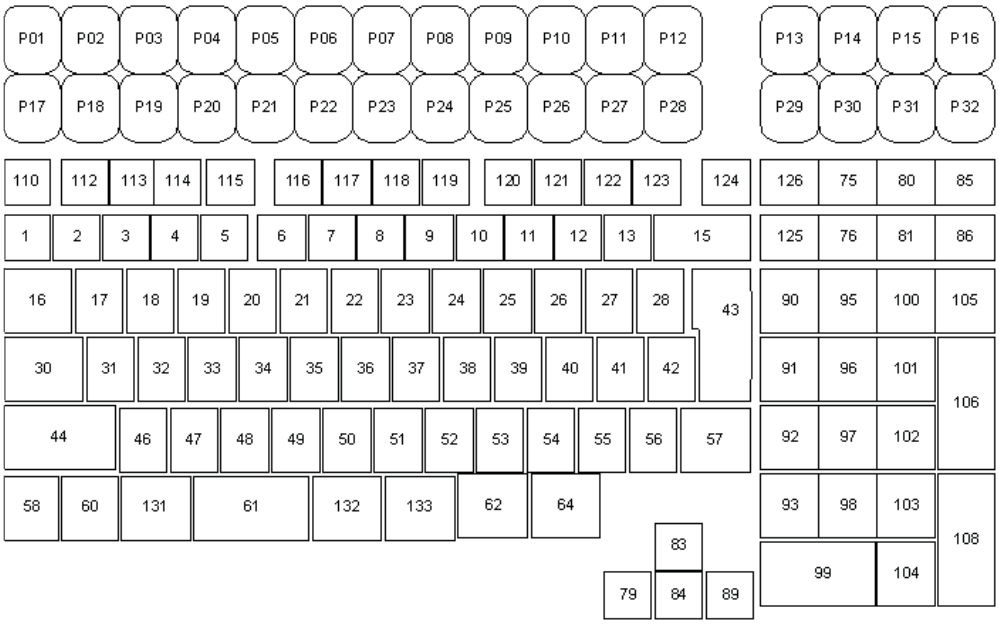
Brazilian Keyboard:



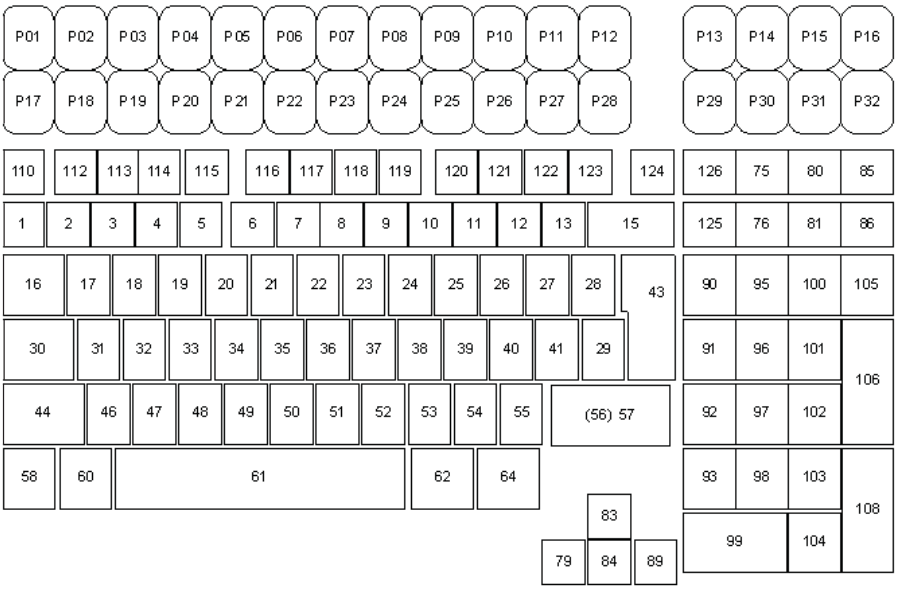
European Keyboard:



Japanese Keyboard:



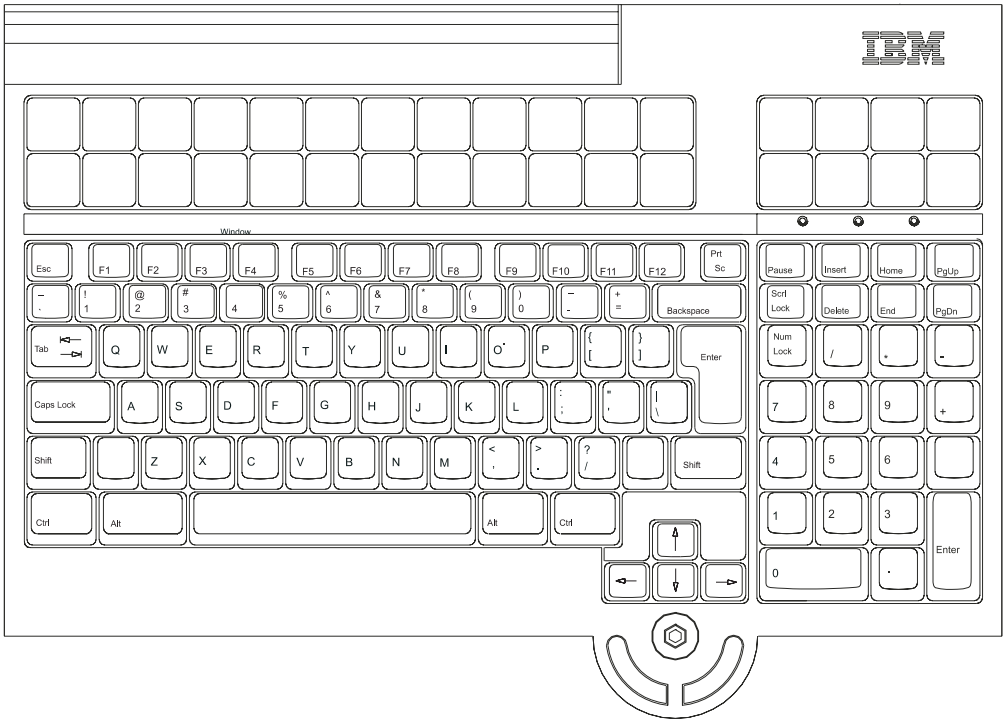
U.S. English Keyboard:



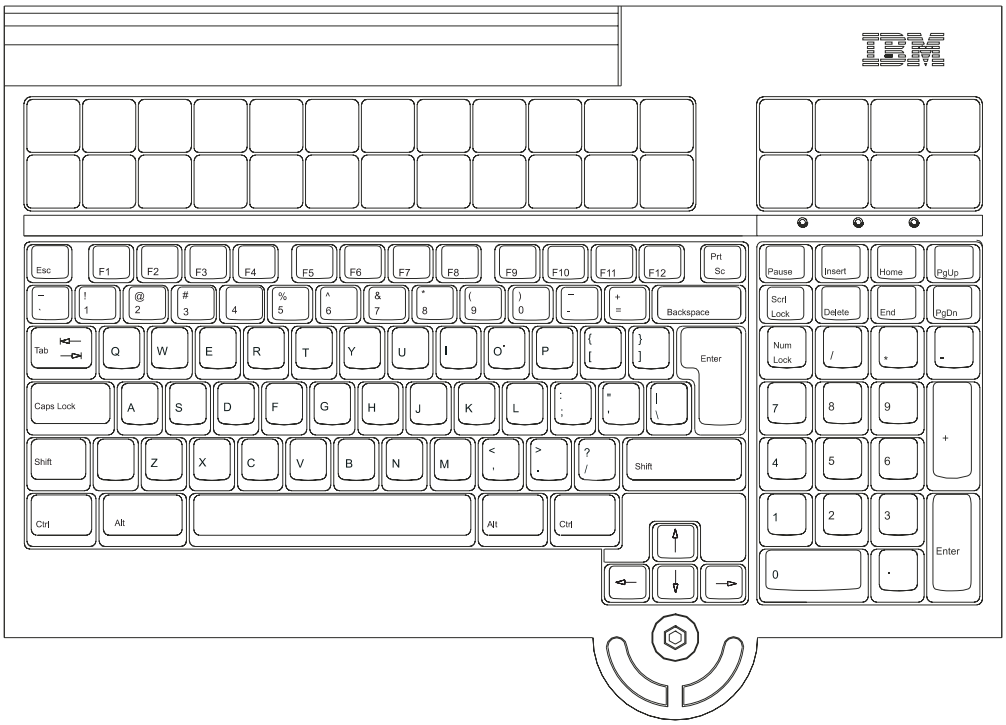
CANPOS Country-Dependent Keyboards

The following pages illustrate the keyboards for all the supported CANPOS keyboards.

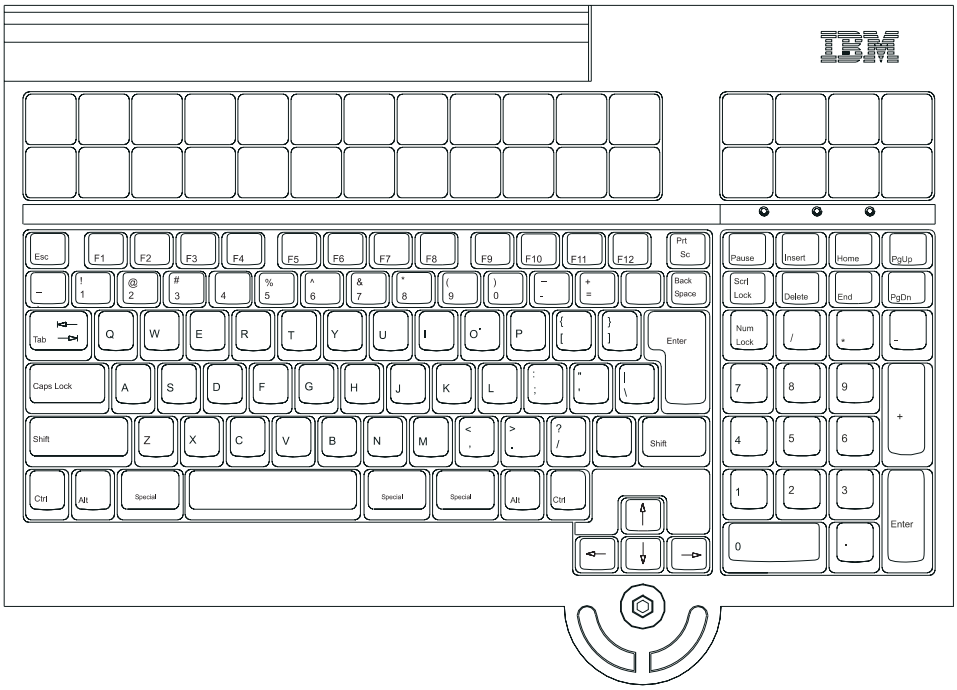
Brazilian Keyboard



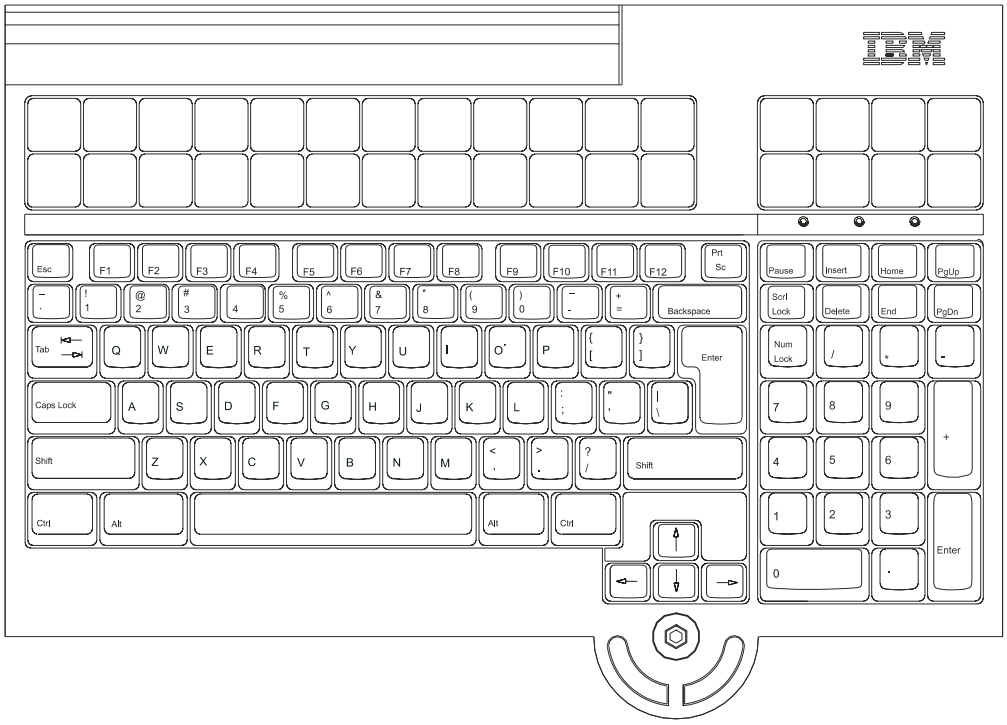
European Keyboard



Japanese Keyboard



U.S. English Keyboard



Chapter 9. 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
- 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:

- 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 306)
- 936 Simplified Chinese
- 949 Korean (see 2 on page 306)
- 950 Traditional Chinese Big5
- 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 2)

Notes:

1. See “Character sets supported by LineDisplay devices” on page 176 for the list of linedisplays and their suported 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 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 62. 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	委	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	蕪	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	景	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 63. 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	秦	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 64. 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	募	墓
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 65. 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	(SH̄Y) 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	< SP030000	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	> SP050000	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	{	I	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	< SA030000	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	> SA050000	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	(SH̄Y) 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	Pts 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	٪ SM020007	5 ND050000	E LE020000	U LU020000	e LE010000	u LU010000	─ SF100000	¼ NF040000	ـأ 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		٧ ND070001	ا AA010000	ط AT450000	هـ AH010003	غ AG310004
-8	╣ SF230000	↑ SM320000	(SP060000	8 ND080000	H LH020000	X LX020000	h LH010000	x LX010000	┤ SF090000	» SP180000	ـا 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	ش AT470003	! SM650000	ض AD450000	ل AL010000
-C	╝ SF250000	∟ SA420000	, SP080000	< SA030000	L LL020000	\ SM070000	l LL010000	| SM130000	┐ SF030000		، SP080007	س AS010000	ج AG230003	¬ SM660000	ع AC470004	ك AK010000
-D	╗ SF390000	↔ SM780000	- SP100000	= SA040000	M LM020000	] SM080000	m LM010000	} SM140000	┌ SF010000	لا AL020000	ج AG230000	ش AS230000	ح AH450003	÷ SA060000	غ 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

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		►	(SP)	0	@	P	`	p	А	Р	а	╗	╝	╜	р	Ё
-1	☺	◄	!	1	A	Q	a	q	Б	С	б	╘	╛	╞	с	ё
-2	☻	↕	"	2	B	R	b	r	В	Т	в	╙	┴	╟	т	Є
-3	♥	!!	#	3	C	S	c	s	Г	У	г	╚	┬	╠	у	є
-4	♦	¶	$	4	D	T	d	t	Д	Ф	д	╔	├	╡	ф	Ї
-5	♣	§	%	5	E	U	e	u	Е	Х	е	╩	─	╢	х	ї
-6	♠	▬	&	6	F	V	f	v	Ж	Ц	ж	╦	╞	╖	ц	Ў
-7	•	↨	'	7	G	W	g	w	З	Ч	з	╠	╟	╕	ч	ў
-8	◘	↑	(	8	H	X	h	x	И	Ш	и	╣	╚	╒	ш	°
-9	○	↓	)	9	I	Y	i	y	Й	Щ	й	╬	╔	╓	щ	∙
-A	◙	→	*	:	J	Z	j	z	К	Ъ	к	╩	╦	╗	ъ	·
-B	♂	←	+	;	K	[	k	{	Л	Ы	л	╗	╦	╖	ы	√
-C	♀	∟	,	<	L	\	l	|	М	Ь	м	╝	╠	╜	ь	№
-D	♪	↔	-	=	M	]	m	}	Н	Э	н	╜	═	╛	э	€
-E	♫	▲	.	>	N	^	n	~	О	Ю	о	╝	╬	╘	ю	■
-F	☼	▼	/	?	O	_	o	⌂	П	Я	п	╒	╩	─	я	(RSP)

Code Page 00866

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	ΐ	▒	╩	Υ	η	±
-2	☻	↕	"	2	B	R	b	r		Ό	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	Ά	Ϋ	Γ	Λ	Π	α	μ	ψ
-7	•	↨	'	7	G	W	g	w	€	©	Δ	Μ	Ρ	β	ν	΄΄
-8	◘	↑	(	8	H	X	h	x	·	Ώ	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	}	Έ	έ	Ι	Ξ	═	δ	ς	ώ
-E	♫	▲	.	>	N	^	n	~	—	ή	«	Ο	╬	ε	τ	■
-F	☼	▼	/	?	O	_	o	⌂	Ή	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 66. 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 67. 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	員	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	村	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	終	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 68. 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	교	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	규	균	귤	그	극	근	귿	글	긁	금	급	긋	긍	긔	기
B1E3	B1E4	B1E5	B1E6	B1E7	B1E8	B1E9	B1EA	B1EB	B1EC	B1ED	B1EE	B1EF	B1F0	B1F1	B1F2
긱	긴	긷	길	긺	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
껨	껩	껭	껴	껸	껼	껿	꺼	꺽	꺾	꼀	꼬	꼭	꼰	꼲	꼴
B2C5	B2C6	B2C7	B2C8	B2C9	B2CA	B2CB	B2CC	B2CD	B2CE	B2CF	B2D0	B2D1	B2D2	B2D3	B2D4
꼼	꼽	꼿	꽁	꽂	꽃	과	곽	괄	괌	괍	괘	괙	괜	괟	괠
B2D5	B2D6	B2D7	B2D8	B2D9	B2DA	B2DB	B2DC	B2DD	B2DE	B2DF	B2E0	B2E1	B2E2	B2E3	B2E4
괌	괍	괭	교	구	국	군	굴	굽	굼	굽	굿	궁	궂	귀	귈

Figure 69. Hangeul (1 of 10)

B2E5	B2E6	B2E7	B2E8	B2E9	B2EA	B2EB	B2EC	B2ED	B2EE	B2EF	B2F0	B2F1	B2F2	B2F3	B2F4
쮸	쮸	쮜	쮜	쮜	쮤	쮜	쮤	쮜	뀌	쮠	쮭	쮠	쮭	끄	끄
B2F5	B2F6	B2F7	B2F8	B2F9	B2FA	B2FB	B2FC	B2FD	B2FE	B3A1	B3A2	B3A3	B3A4	B3A5	B3A6
끄	끄	끄	끄	끄	끄	끄	끄	끄	끄	끄	끼	끼	끼	끼	끼
B3A7	B3A8	B3A9	B3AA	B3AB	B3AC	B3AD	B3AE	B3AF	B3B0	B3B1	B3B2	B3B3	B3B4	B3B5	B3B6
끼	끼	끼	나	나	나	나	나	낚	낚	낚	나	나	나	나	나
B3B7	B3B8	B3B9	B3BA	B3BB	B3BC	B3BD	B3BE	B3BF	B3C0	B3C1	B3C2	B3C3	B3C4	B3C5	B3C6
나	나	낟	낭	내	내	내	낼	내	낼	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	넋	넋	너	너	너	너	너	넝
B3D7	B3D8	B3D9	B3DA	B3DB	B3DC	B3DD	B3DE	B3DF	B3E0	B3E1	B3E2	B3E3	B3E4	B3E5	B3E6
네	네	네	넨	네	넨	네	넨	네	너	녀	녀	녀	녀	녀	녀
B3E7	B3E8	B3E9	B3EA	B3EB	B3EC	B3ED	B3EE	B3EF	B3F0	B3F1	B3F2	B3F3	B3F4	B3F5	B3F6
녀	녀	네	넨	노	녹	논	놀	놂	놈	놉	노	농	높	놓	나
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	뉬	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	늙	늚	늠	늡	늣
B4C9	B4CA	B4CB	B4CC	B4CD	B4CE	B4CF	B4D0	B4D1	B4D2	B4D3	B4D4	B4D5	B4D6	B4D7	B4D8
능	늑	늘	니	닌	닐	니	니	니	닐	닒	니	닙	니	니	니
B4D9	B4DA	B4DB	B4DC	B4DD	B4DE	B4DF	B4E0	B4E1	B4E2	B4E3	B4E4	B4E5	B4E6	B4E7	B4E8
다	닥	닦	단	닫	달	닭	닮	닯	닳	담	답	닷	닸	당	닥
B4E9	B4EA	B4EB	B4EC	B4ED	B4EE	B4EF	B4F0	B4F1	B4F2	B4F3	B4F4	B4F5	B4F6	B4F7	B4F8
닦	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
더	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	뎌	뎐	뎔	뎠	뎡	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	돕	돗	동	돛	돝	돠	돤	돨	돼	됐	되	된	될	됨	

Figure 70. Hangeul (2 of 10)

B5CB	B5CC	B5CD	B5CE	B5CF	B5D0	B5D1	B5D2	B5D3	B5D4	B5D5	B5D6	B5D7	B5D8	B5D9	B5DA
뒵	뒤	듀	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	듑	드	득	든	들	글	듚
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	떩	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
똴	뙈	뙤	뙨	뚀	뚜	뚝	뚠	뚤	뚬	뚱	뒈	뒤	뒨	뒬	뒴
B6DD	B6DE	B6DF	B6E0	B6E1	B6E2	B6E3	B6E4	B6E5	B6E6	B6E7	B6E8	B6E9	B6EA	B6EB	B6EC
뒵	뒹	뜨	뜩	뜬	뜰	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	라
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	렇	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
뢰	뢰	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	뤰	뤳	뤵	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	릌	릔	릘	릠	릡	릣	릥	릦	릧	릩	리	릭	린

Figure 71. Hangeul (3 of 10)

B8B1	B8B2	B8B3	B8B4	B8B5	B8B6	B8B7	B8B8	B8B9	B8BA	B8BB	B8BC	B8BD	B8BE	B8BF	B8C0
맇	맊	맋	맛	맣	매	맥	맨	맩	맪	맫	맬	맭	맮	맯	맰
B8C1	B8C2	B8C3	B8C4	B8C5	B8C6	B8C7	B8C8	B8C9	B8CA	B8CB	B8CC	B8CD	B8CE	B8CF	B8D0
맱	맲	맳	맴	맵	맶	맷	맸	맹	맺	맻	맼	맽	맾	맿	먀
B8D1	B8D2	B8D3	B8D4	B8D5	B8D6	B8D7	B8D8	B8D9	B8DA	B8DB	B8DC	B8DD	B8DE	B8DF	B8E0
먁	먂	머	먹	먺	먻	먼	먽	먾	먿	멁	멂	멃	멄	멆	멇
B8E1	B8E2	B8E3	B8E4	B8E5	B8E6	B8E7	B8E8	B8E9	B8EA	B8EB	B8EC	B8ED	B8EE	B8EF	B8F0
멈	멉	멋	멌	멍	멎	멓	메	멕	멖	멗	멘	멚	멛	멜	멝
B8F1	B8F2	B8F3	B8F4	B8F5	B8F6	B8F7	B8F8	B8F9	B8FA	B8FB	B8FC	B8FD	B8FE	B9A1	B9A2
멞	멟	멠	멡	멢	멣	멤	멥	멦	멧	멨	멩	멪	멫	메	멮
B9A3	B9A4	B9A5	B9A6	B9A7	B9A8	B9A9	B9AA	B9AB	B9AC	B9AD	B9AE	B9AF	B9B0	B9B1	B9B2
멯	며	멱	멲	멳	면	멵	멶	멷	멸	멹	멺	멻	명	몀	몁
B9B3	B9B4	B9B5	B9B6	B9B7	B9B8	B9B9	B9BA	B9BB	B9BC	B9BD	B9BE	B9BF	B9C0	B9C1	B9C2
몂	몃	몄	명	몆	몇	몈	몉	몊	몋	몌	몍	몎	몏	몐	몑
B9C3	B9C4	B9C5	B9C6	B9C7	B9C8	B9C9	B9CA	B9CB	B9CC	B9CD	B9CE	B9CF	B9D0	B9D1	B9D2
몒	몓	몔	몕	몖	몗	몘	몙	몚	몛	몜	몝	몞	몟	몠	몡
B9D3	B9D4	B9D5	B9D6	B9D7	B9D8	B9D9	B9DA	B9DB	B9DC	B9DD	B9DE	B9DF	B9E0	B9E1	B9E2
몢	몣	몤	몥	몦	몧	바	박	밖	밗	반	받	발	밠	밡	밢
B9E3	B9E4	B9E5	B9E6	B9E7	B9E8	B9E9	B9EA	B9EB	B9EC	B9ED	B9EE	B9EF	B9F0	B9F1	B9F2
밣	밤	밥	밦	밧	배	백	밪	밫	밬	밭	밮	밯	배	백	뱌
B9F3	B9F4	B9F5	B9F6	B9F7	B9F8	B9F9	B9FA	B9FB	B9FC	B9FD	B9FE	BAA1	BAA2	BAA3	BAA4
뱍	뱎	뱏	버	벅	번	벋	벌	벍	벎	벏	벐	벑	벒	베	벖
BAA5	BAA6	BAA7	BAA8	BAA9	BAAA	BAAB	BAAC	BAAD	BAAE	BAAF	BAB0	BAB1	BAB2	BAB3	BAB4
벗	벘	벙	벚	벛	벜	벝	벞	벟	베	벡	벢	벣	벤	벥	벦
BAB5	BAB6	BAB7	BAB8	BAB9	BABA	BABB	BABC	BABD	BABE	BABF	BAC0	BAC1	BAC2	BAC3	BAC4
벧	벩	벨	벫	벬	벭	벮	벯	벰	벱	벲	벳	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	뵫	뵬	뵭	뵮	뵯	뵰	뵱
BAD5	BAD6	BAD7	BAD8	BAD9	BADA	BADB	BADC	BADD	BADE	BADF	BAE0	BAE1	BAE2	BAE3	BAE4
뵲	뵳	뵴	뵵	뵶	뵷	뵸	뵹	뵺	뵻	뵼	뵽	뵾	뵿	부	북
BAE5	BAE6	BAE7	BAE8	BAE9	BAEA	BAEB	BAEC	BAED	BAEE	BAEF	BAF0	BAF1	BAF2	BAF3	BAF4
분	불	붐	붑	붕	붖	붗	붘	붙	붚	붛	붜	붝	붞	붟	붡

Figure 72. Hangeul (4 of 10)

BAF5	BAF6	BAF7	BAF8	BAF9	BAFA	BAFB	BAFC	BAFD	BAFE	BBA1	BBA2	BBA3	BBA4	BBA5	BBA6
빙	빙	빙	빙	빙	빙	빙	빠	빡	빤	빨	빠	빠	빱	빠	빴
BBA7	BBA8	BBA9	BBAA	BBAB	BBAC	BBAD	BBAE	BBAF	BBB0	BBB1	BBB2	BBB3	BBB4	BBB5	BBB6
빠	빵	빼	빽	빼	빨	빼	빱	빼	빴	빼	빠	빡	빡	뻐	뻑
BBB7	BBB8	BBB9	BBBA	BBBB	BBBC	BBBD	BBBE	BBBF	BBC0	BBC1	BBC2	BBC3	BBC4	BBC5	BBC6
뻐	뻐	뻑	뻐	뻐	뻐	뻐	뻬	뻥	뻐	뻑	뻑	뻑	뻑	뻑	뻑
BBC7	BBC8	BBC9	BBCA	BBCB	BBCC	BBCD	BBCE	BBCF	BBD0	BBD1	BBD2	BBD3	BBD4	BBD5	BBD6
뽀	뽁	뽄	뽈	뽐	뽑	뽕	삐	뽀	뽕	뿌	뿍	뿐	뿔	뿜	쁘
BBD7	BBD8	BBD9	BBDA	BBDB	BBDC	BBDD	BBDE	BBDF	BBE0	BBE1	BBE2	BBE3	BBE4	BBE5	BBE6
뿡	뿝	뿡	쁘	쁜	쁠	쁨	쁩	삐	삐	삐	삘	삐	삡	삐	삐
BBE7	BBE8	BBE9	BBEA	BBEB	BBEC	BBED	BBEE	BBEF	BBF0	BBF1	BBF2	BBF3	BBF4	BBF5	BBF6
사	삭	삯	산	삔	삘	삶	삷	삼	삽	삿	삿	상	삳	새	색
BBF7	BBF8	BBF9	BBFA	BBFB	BBFC	BBFD	BBFE	BCA1	BCA2	BCA3	BCA4	BCA5	BCA6	BCA7	BCA8
샌	샐	샘	샙	샛	샜	생	샤	샥	샨	샬	샴	샵	샷	샹	섀
BCA9	BCAA	BCAB	BCAC	BCAD	BCAE	BCAF	BCB0	BCB1	BCB2	BCB3	BCB4	BCB5	BCB6	BCB7	BCB8
섄	섈	섬	섕	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	섹	섄	섈	섬	섭	섯	섰	성	셔	셕	셔	셜
BCC9	BCCA	BCCB	BCCC	BCCD	BCCE	BCCF	BCD0	BCD1	BCD2	BCD3	BCD4	BCD5	BCD6	BCD7	BCD8
셤	셥	셧	셨	셩	셰	셴	셸	셩	소	속	솎	손	솔	솖	솜
BCD9	BCDA	BCDB	BCDC	BCDD	BCDE	BCDF	BCE0	BCE1	BCE2	BCE3	BCE4	BCE5	BCE6	BCE7	BCE8
솝	솟	송	솥	쇠	쇡	쇤	쇨	쇵	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	숀	숄	숌	숍	숏	숑	수	쉭	순
BCF9	BCFA	BCFB	BCFC	BCFD	BCFE	BDA1	BDA2	BDA3	BDA4	BDA5	BDA6	BDA7	BDA8	BDA9	BDAA
숟	술	숨	숩	숫	숭	쉬	쉭	쉰	쉴	쉼	쉽	쉿	슁	쉐	쉑
BDAB	BDAC	BDAD	BDAE	BDAF	BDB0	BDB1	BDB2	BDB3	BDB4	BDB5	BDB6	BDB7	BDB8	BDB9	BDBA
쉔	쉬	쉭	쉰	쉴	쉼	쉽	쉿	쉥	슈	슉	슐	슘	슛	슝	스
BDBB	BDBC	BDBD	BDBE	BDBF	BDC0	BDC1	BDC2	BDC3	BDC4	BDC5	BDC6	BDC7	BDC8	BDC9	BDCA
슥	슨	슬	슭	슴	습	슷	슝	시	식	신	실	싫	심	십	싯
BDCB	BDCC	BDCD	BDCE	BDCF	BDD0	BDD1	BDD2	BDD3	BDD4	BDD5	BDD6	BDD7	BDD8	BDD9	BDDA
싱	싱	싶	싸	싹	싼	싼	쌀	쌈	쌉	쌋	쌍	쌓	쌔	쌕	쌘

Figure 73. Hangeul (5 of 10)

BDDB	BDDC	BDDD	BDDE	BDDF	BDE0	BDE1	BDE2	BDE3	BDE4	BDE5	BDE6	BDE7	BDE8	BDE9	BDEA
쌜	쌔	쌥	쌨	쌰	쌳	써	쎄	쎈	쎌	쎔	쎠	쎤	쎴	쏑	쏘
BDEB	BDEC	BDED	BDEE	BDEF	BDF0	BDF1	BDF2	BDF3	BDF4	BDF5	BDF6	BDF7	BDF8	BDF9	BDFA
쏜	쏠	쏜	쏘	쏙	쏜	쏟	쏠	쏢	쏨	쏩	쏭	쏴	쏸	쏸	쏼
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
쒼	쒜	쒸	쒼	쓩	쓰	쓱	쓴	쓸	쓺	쓼	쓽	쓾	씌	씐	씔
BEBD	BEBE	BEBF	BEC0	BEC1	BEC2	BEC3	BEC4	BEC5	BEC6	BEC7	BEC8	BEC9	BECA	BECB	BECC
씜	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	얼	얽	얾	엄	업	없	엇	었	엉	엊	엌
BEFD	BEFE	BFA1	BFA2	BFA3	BFA4	BFA5	BFA6	BFA7	BFA8	BFA9	BFAA	BFAB	BFAC	BFAD	BFAE
엎	엏	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
엷	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	옰	옴	옵	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	았	왕	왜	왝	왠	왤	왬	왭	외	왹	왼
BFDF	BFE0	BFE1	BFE2	BFE3	BFE4	BFE5	BFE6	BFE7	BFE8	BFE9	BFEA	BFEB	BFEC	BFED	BFEE
욀	욈	욉	욋	욍	요	욕	욘	욜	욤	욥	욧	용	우	욱	운
BFEF	BFF0	BFF1	BFF2	BFF3	BFF4	BFF5	BFF6	BFF7	BFF8	BFF9	BFFA	BFFB	BFFC	BFFD	BFFE
울	욿	움	웁	웃	웄	워	웍	원	월	웜	웝	웠	웡	웥	웨
C0A1	C0A2	C0A3	C0A4	C0A5	C0A6	C0A7	C0A8	C0A9	C0AA	C0AB	C0AC	C0AD	C0AE	C0AF	C0B0
웩	웬	웰	웸	웹	웽	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 74. Hangeul (6 of 10)

C0C1	C0C2	C0C3	C0C4	C0C5	C0C6	C0C7	C0C8	C0C9	C0CA	C0CB	C0CC	C0CD	C0CE	C0CF	C0D0
읗	의	읙	읚	읛	읜	읝	읞	읟	읠	읡	읢	읣	읤	읥	읦
C0D1	C0D2	C0D3	C0D4	C0D5	C0D6	C0D7	C0D8	C0D9	C0DA	C0DB	C0DC	C0DD	C0DE	C0DF	C0E0
읧	읨	읩	읪	읫	읬	읭	읮	읯	자	작	잔	잖	잗	잘	잙
C0E1	C0E2	C0E3	C0E4	C0E5	C0E6	C0E7	C0E8	C0E9	C0EA	C0EB	C0EC	C0ED	C0EE	C0EF	C0F0
잚	잛	잜	잝	잞	잟	재	잢	잣	잤	장	잦	잧	잨	잩	챠
C0F1	C0F2	C0F3	C0F4	C0F5	C0F6	C0F7	C0F8	C0F9	C0FA	C0FB	C0FC	C0FD	C0FE	C1A1	C1A2
챡	챢	챣	챤	챥	챦	챧	챨	챩	처	척	천	철	첟	첡	첢
C1A3	C1A4	C1A5	C1A6	C1A7	C1A8	C1A9	C1AA	C1AB	C1AC	C1AD	C1AE	C1AF	C1B0	C1B1	C1B2
첣	첤	첥	체	첧	첨	첩	첪	첫	첬	청	쳐	쳔	쳕	쳖	쳗
C1B3	C1B4	C1B5	C1B6	C1B7	C1B8	C1B9	C1BA	C1BB	C1BC	C1BD	C1BE	C1BF	C1C0	C1C1	C1C2
쳘	쳙	쳚	쳛	쳜	쳝	쳞	쳟	쳠	쳡	쳢	쳣	쳤	쳥	쳦	초
C1C3	C1C4	C1C5	C1C6	C1C7	C1C8	C1C9	C1CA	C1CB	C1CC	C1CD	C1CE	C1CF	C1D0	C1D1	C1D2
촉	촐	촙	촛	총	촤	촥	촦	촧	촨	촩	촪	촫	촬	촭	촮
C1D3	C1D4	C1D5	C1D6	C1D7	C1D8	C1D9	C1DA	C1DB	C1DC	C1DD	C1DE	C1DF	C1E0	C1E1	C1E2
쵸	쵹	쵺	쵻	쵼	쵽	쵾	쵿	최	쵠	쵡	쵢	쵣	쵤	쵥	쵦
C1E3	C1E4	C1E5	C1E6	C1E7	C1E8	C1E9	C1EA	C1EB	C1EC	C1ED	C1EE	C1EF	C1F0	C1F1	C1F2
쵧	쵨	쵩	쵪	쵫	쵬	쵭	쵮	쵯	쵰	쵱	쵲	쵳	쵴	쵵	쵶
C1F3	C1F4	C1F5	C1F6	C1F7	C1F8	C1F9	C1FA	C1FB	C1FC	C1FD	C1FE	C2A1	C2A2	C2A3	C2A4
쵷	쵸	쵹	지	직	진	짇	질	짊	짋	짌	짍	짎	짏	짐	집
C2A5	C2A6	C2A7	C2A8	C2A9	C2AA	C2AB	C2AC	C2AD	C2AE	C2AF	C2B0	C2B1	C2B2	C2B3	C2B4
짜	짝	짠	짢	짣	짤	짥	짦	짧	짨	짩	째	짬	짭	짮	짯
C2B5	C2B6	C2B7	C2B8	C2B9	C2BA	C2BB	C2BC	C2BD	C2BE	C2BF	C2C0	C2C1	C2C2	C2C3	C2C4
짱	짲	짳	짴	짵	쨔	쨕	쨖	쩌	쩍	쩐	쩔	쩜	쩝	쩟	쩠
C2C5	C2C6	C2C7	C2C8	C2C9	C2CA	C2CB	C2CC	C2CD	C2CE	C2CF	C2D0	C2D1	C2D2	C2D3	C2D4
쩡	쩢	쩣	쩤	쩥	쪼	쪽	쪾	쪿	쫀	쫄	쫌	쫍	쫏	쫑	쫒
C2D5	C2D6	C2D7	C2D8	C2D9	C2DA	C2DB	C2DC	C2DD	C2DE	C2DF	C2E0	C2E1	C2E2	C2E3	C2E4
쫓	쫔	쫕	쫖	쫗	쫘	쫙	쫚	쫛	쫜	쫝	쫞	쫟	쫠	쫡	쫢
C2E5	C2E6	C2E7	C2E8	C2E9	C2EA	C2EB	C2EC	C2ED	C2EE	C2EF	C2F0	C2F1	C2F2	C2F3	C2F4
쫣	쫤	쫥	쫦	쫧	쫨	쫩	쫪	쫫	쫬	쫭	쫮	쫯	쫰	쫱	쫲
C2F5	C2F6	C2F7	C2F8	C2F9	C2FA	C2FB	C2FC	C2FD	C2FE	C3A1	C3A2	C3A3	C3A4	C3A5	C3A6
쫳	쫴	차	착	찬	찮	찰	참	찹	찻	찼	창	찾	채	책	챈

Figure 75. 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	처	척	천	철
C3B7	C3B8	C3B9	C3BA	C3BB	C3BC	C3BD	C3BE	C3BF	C3C0	C3C1	C3C2	C3C3	C3C4	C3C5	C3C6
첩	첩	첫	첬	청	체	첵	첸	첼	쳄	쳅	쳇	쳉	쳐	쳔	쳤
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	최	촨	촬	촹	쵀
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
췌	췐	췔	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	퀌	퀍	퀏	퀴	큐	큭	큰	큘	큼	큽	크
C5AB	C5AC	C5AD	C5AE	C5AF	C5B0	C5B1	C5B2	C5B3	C5B4	C5B5	C5B6	C5B7	C5B8	C5B9	C5BA
큰	클	큼	큽	킁	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 76. Hangeul (8 of 10)

C5EB	C5EC	C5ED	C5EE	C5EF	C5F0	C5F1	C5F2	C5F3	C5F4	C5F5	C5F6	C5F7	C5F8	C5F9	C5FA
퓽	퓾	탸	퍈	턔	틔	틘	틙	틚	틛	틝	퉈	퉉	퉋	퉍	퉏
C5FB	C5FC	C5FD	C5FE	C6A1	C6A2	C6A3	C6A4	C6A5	C6A6	C6A7	C6A8	C6A9	C6AA	C6AB	C6AC
퉑	퉒	퉓	퉕	퉤	퉧	퉩	퉪	퉫	퉭	퉯	퉱	퉳	퉵	퉷	퉹
C6AD	C6AE	C6AF	C6B0	C6B1	C6B2	C6B3	C6B4	C6B5	C6B6	C6B7	C6B8	C6B9	C6BA	C6BB	C6BC
퉻	퉽	퉿	튀	튂	튄	튆	튇	튈	튉	튋	튍	튎	튐	튒	튓
C6BD	C6BE	C6BF	C6C0	C6C1	C6C2	C6C3	C6C4	C6C5	C6C6	C6C7	C6C8	C6C9	C6CA	C6CB	C6CC
튔	튕	튖	튗	튘	튙	튚	파	팍	팎	판	팒	팓	팔	팕	팖
C6CD	C6CE	C6CF	C6D0	C6D1	C6D2	C6D3	C6D4	C6D5	C6D6	C6D7	C6D8	C6D9	C6DA	C6DB	C6DC
팗	팘	팙	패	팩	팪	팫	팬	팭	팮	팯	팰	팱	퍄	퍅	퍼
C6DD	C6DE	C6DF	C6E0	C6E1	C6E2	C6E3	C6E4	C6E5	C6E6	C6E7	C6E8	C6E9	C6EA	C6EB	C6EC
퍽	퍾	퍿	펁	펂	펃	펄	페	펙	펚	펛	펜	펝	펞	펟	펠
C6ED	C6EE	C6EF	C6F0	C6F1	C6F2	C6F3	C6F4	C6F5	C6F6	C6F7	C6F8	C6F9	C6FA	C6FB	C6FC
펡	펢	펣	펤	펥	펦	펧	펨	펩	펪	펫	펬	펭	펮	펯	펰
C6FD	C6FE	C7A1	C7A2	C7A3	C7A4	C7A5	C7A6	C7A7	C7A8	C7A9	C7AA	C7AB	C7AC	C7AD	C7AE
펱	펲	퍄	퍅	퍼	퍽	퍾	퍿	펁	펂	펃	펄	펅	펆	펇	펈
C7AF	C7B0	C7B1	C7B2	C7B3	C7B4	C7B5	C7B6	C7B7	C7B8	C7B9	C7BA	C7BB	C7BC	C7BD	C7BE
펉	펊	펋	펌	펍	펎	펏	펐	페	펙	펚	펛	펜	펝	펞	펟
C7BF	C7C0	C7C1	C7C2	C7C3	C7C4	C7C5	C7C6	C7C7	C7C8	C7C9	C7CA	C7CB	C7CC	C7CD	C7CE
펠	펡	펢	펣	펤	펥	펦	펧	펨	펩	펪	펫	펬	펭	펮	펯
C7CF	C7D0	C7D1	C7D2	C7D3	C7D4	C7D5	C7D6	C7D7	C7D8	C7D9	C7DA	C7DB	C7DC	C7DD	C7DE
하	학	한	할	핥	함	합	핫	항	해	핵	핸	핼	핾	핿	햇
C7DF	C7E0	C7E1	C7E2	C7E3	C7E4	C7E5	C7E6	C7E7	C7E8	C7E9	C7EA	C7EB	C7EC	C7ED	C7EE
행	향	햐	향	허	헉	헌	헐	헒	헓	헔	헕	헖	헤	헥	헨
C7EF	C7F0	C7F1	C7F2	C7F3	C7F4	C7F5	C7F6	C7F7	C7F8	C7F9	C7FA	C7FB	C7FC	C7FD	C7FE
헬	헴	헵	헷	헹	혀	혁	현	혈	혐	협	혓	혔	형	혜	혠
C8A1	C8A2	C8A3	C8A4	C8A5	C8A6	C8A7	C8A8	C8A9	C8AA	C8AB	C8AC	C8AD	C8AE	C8AF	C8B0
혤	혭	호	혹	혼	홀	홅	홈	홉	홋	홍	홑	화	확	환	활
C8B1	C8B2	C8B3	C8B4	C8B5	C8B6	C8B7	C8B8	C8B9	C8BA	C8BB	C8BC	C8BD	C8BE	C8BF	C8C0
홧	황	홰	횃	횐	횔	횕	회	획	획	횟	횡	횢	횟	효	횬
C8C1	C8C2	C8C3	C8C4	C8C5	C8C6	C8C7	C8C8	C8C9	C8CA	C8CB	C8CC	C8CD	C8CE	C8CF	C8D0
횰	횹	횻	후	훅	훈	훌	훑	훔	훗	훙	훠	훤	훨	훰	훵

Figure 77. Hangeul (9 of 10)

C8D1	C8D2	C8D3	C8D4	C8D5	C8D6	C8D7	C8D8	C8D9	C8DA	C8DB	C8DC	C8DD	C8DE	C8DF	C8E0
퀘	퀘	퀘	퀘	퀘	퀴	퀴	퀴	퀴	퀴	퀴	퀴	퀴	큐	큐	큐

C8E1	C8E2	C8E3	C8E4	C8E5	C8E6	C8E7	C8E8	C8E9	C8EA	C8EB	C8EC	C8ED	C8EE	C8EF	C8F0
큘	큠	큠	큠	큐	큰	큰	큰	큰	큰	큰	큰	큰	큰	큰	큰

C8F1	C8F2	C8F3	C8F4	C8F5	C8F6	C8F7	C8F8	C8F9	C8FA	C8FB	C8FC	C8FD	C8FE
킈	킨	킬	킴	킵	킹	히	힉	힌	힐	힘	힙	힛	힝

Figure 78. 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 196. 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)	EIA232/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.

Cash Drawer

Table 197. Cash Drawer

System/device	Bus	Model name value	
		A	B
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	Embedded	4614/4615 - A	4614/4615 - B

Table 198. 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 Embed-ded
UnifiedPOSVersion	Y	Y	Y	Y	Y	Y	Y	Y
DeviceCategory	Y	Y	Y	Y	Y	Y	Y	Y
ManufacturerName	Y	Y	Y	Y	Y	Y	Y	Y
ModelName	Y	Y	Y	Y	Y	Y	Y	Y
SerialNumber	Y	Y	N	N	N	N	N	N
ManufactureDate	N	N	N	N	N	N	N	N
Mechanical Revision	N	N	N	N	N	N	N	N
FirmwareRevision	Y	Y	N	N	N	N	N	N
Interface	Y	Y	Y	Y	Y	Y	Y	Y
InstallationDate	N	N	N	N	N	N	N	N
HoursPowered Count	N	N	N	N	N	N	N	N
Communication ErrorCount	N	N	N	N	N	N	N	N

Table 199. 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 Embed-ded
DrawerGoodOpen Count	N	N	N	N	N	N	N	N
DrawerFailedOpen Count	N	N	N	N	N	N	N	N

Note: 730/750 USB devices use BCD level for Firmware Revision.

Check Scanner

Table 200. Check Scanner

Device	Bus	Model name value
POSPrinter 4610-TI8/TI9	USB / EIA232 / RS485	4610

Table 201. 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 202. Check Scanner manufacturer properties

Manufacturer Properties	unitofmeasure	Description	POSPrinter 4610 -TI8/TI9 - EIA232/RS485/USB Supported:
IBM_CheckScannedCount	”” - empty	Number of checks scanned	Y

Table 202. Check Scanner manufacturer properties (continued)

Manufacturer Properties	unitofmeasure	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 (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 203. 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 204. 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 205. 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 206. 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 207. 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 208. 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 208. 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 Keylock	Embedded	4614/4615
Keyboard 4685-K03	USB	4685-K03
Keyboard 4685-K03	RS-485	4674/4685-K03

Table 209. 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	N	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	N	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

Table 210. Keylock specific properties

Specific Properties	Description	USB Supported	RS485 Supported	PS2 Supported	Embedded
LockPositionChangeCount	Number of lock position changes	N	N	N	N

Note: USB devices use BCD level for Firmware Revision.

Line Display

Table 211. 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

Table 211. Line Display (continued)

Device	Bus	Model name value
2X20 LCD	USB/RS485	2X20 LCD
APA Display (SBCS/Japan/Korea/....)	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-PLU

Table 212. 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 213. 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 214. MICR

Device	Bus	Model Name value
PosPrinter 4610-TI4/5/8/9	EIA232/RS485/USB	4610

Table 215. MICR common properties

Common Properties	Description	EIA232/RS485/USB Supported
UnifiedPOSVersion	Version of the UnifiedPOS specification supported	Y
DeviceCategory	Device category	Y

Table 215. MICR common properties (continued)

Common Properties	Description	EIA232/RS485/USB Supported
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 216. 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

Notes:

1. POSPrinter microcode level used for Firmware Revision.
2. ManufacturerDate is only available for newer POSPrinters (Dec., 2005). Older ones are not supported.
3. DeviceCategory is value is taken from the programmatic name.

Motion Sensor

Table 217. Motion Sensor

Device	Bus	Model Name Value
SurePOS 500/600/Kiosk	Embedded	IBM Generic Motion Sensor

Table 218. 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

Table 218. Motion Sensor common properties (continued)

Common Properties	Description	Embedded Supported
Interface	Device hardware interface	Y
InstallationDate	Device installation date	N
HoursPoweredCount	Number of hours powerd On	N
CommunicationErrorCount	Number of communication errors	N

Table 219. Motion Sensor specific properties

Specific Properties	Description	Supported
MotionEventCount	Number of motion occurrences	N

MSR

Table 220. MSR

Device	Bus	Model Name value
NANPOS ISO MSR	USB	NANPOS - ISO
NANPOS as System attached ISO MSR	PS2/USB	NANPOS - ISO
4820 ISO MSR (SurePoint)	USB	4820 - ISO
4820 JUCC MSR (SurePoint)	USB	4820 - JUCC
4820/4835/4838/4840/4851 ISO MSR 500/600/Kiosk	EIA232	4820/4835/4838/4840/4851 - ISO
4820/4835/4840/4851 JUCC MSR 500/600/Kiosk	EIA232	4820/4835/4840/4851 - JUCC
50-key ISO MSR	USB	50-key - ISO
CANPOS ISO MSR	PS2	CANPOS - ISO
ANKPOS JUCC MSR	USB	ANKPOS - JUCC
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 ISO MSR	PS2	4614/4615 - ISO
50-key JUCC MSR	USB	50-key - JUCC
Keyboard 4685 K03 JUCC MSR	USB	4685-K03 - JUCC

Table 220. MSR (continued)

Device	Bus	Model Name value
Keyboard 4685 K03 JUCC MSR	RS485	4685 / Kbd V - JUCC

Table 221. 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	N	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	N	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

Table 222. MSR specific properties

Specific Properties	Description	USB Supported	PS2 Supported	EIA232 Supported	RS485 Supported
GoodReadCount	Number of successful reads	N	N	N	N
FailedReadCount	Number of failed reads	Y	N	N	N
UnreadableCardCount	Number of failed data parses	N	N	N	N

Notes:

1. USB devices use BCD level for Firmware Revision.
2. Device category value is taken from programmatic name.

POSKeyboard

Table 223. POSKeyboard

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

Table 223. POSKeyboard (continued)

Device	Bus	Model Name Value
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 Keyboard	PS2	4614/4615
50-key POSKeyboards JUCC MSR - DBCS	RS485/USB	50-key
Keyboard 4685 K03	USB	4685-K03
Keyboard 4685 K03	RS485	4674/4685-K03

Table 224. 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	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 225. POS Keyboard specific properties

Specific Properties	Description	USB Supported	PS2 Supported	RS485 Supported
KeyPressedCount	Number of keys pressed	N	N	N

Note: USB devices use BCD level for Firmware Revision.

POSPrinter

Table 226. POSPrinter

Device	Bus	Model Name value
4610 TI3 (thermal/impact)	EIA232/RS485/USB	4610-TI3/4
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 Thermal	EIA232	4614/4615

Table 227. POSPrinter common properties

Common properties	Description	4610 Printer USB/EIA232/ RS485 Supported	4674/4689 RS485	4689 USB	SureONE 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 228. POSPrinter 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 doucment/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	N	N	N
PrinterFaultCount	Number of printer faults	N	N	N	N
PrintSideChangeCount	Number of print side changes (or check flips) performed	Y*	N	N	N
FailedPrintSideChangeCount	Number of print side change (of check flip) failures	Y*	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

* Only 4610-TI4/8/9 models.

Notes:

1. POSPrinter microcode level used for Firmware Revision.
2. ManufacturerDate is only available for new POSPrinters (Dec., 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.

Scale

Table 229. 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 230. Scale common properties

Common Properties	Description	USB Supported	RS485 Supported
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

Table 231. 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 232. 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

Table 232. Scanner (continued)

Device	Bus	Model Name value	Manufacturer
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
Kiosk - 4838 Line	EIA232	4838-Line	OEM
Kiosk - 4838 Omni	EIA232	4838-Omni	OEM

Table 233. 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 234. 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 235. 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
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

Table 236. 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	N	N	N	N
MechanicalRevision	Device hardware revision	N	N	N	N

Table 236. Tone Indicator common properties (continued)

Common Properties	Description	4610 Tx6/Tx7 Printer Tone Indicator	USB Supported	PS2 Supported	RS485 Supported
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	N	N	N	N
CommunicationErrorCount	Number of communication errors	N	N	N	N

Table 237. 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

At startup, after the splash screen, the first dialog appears as shown:

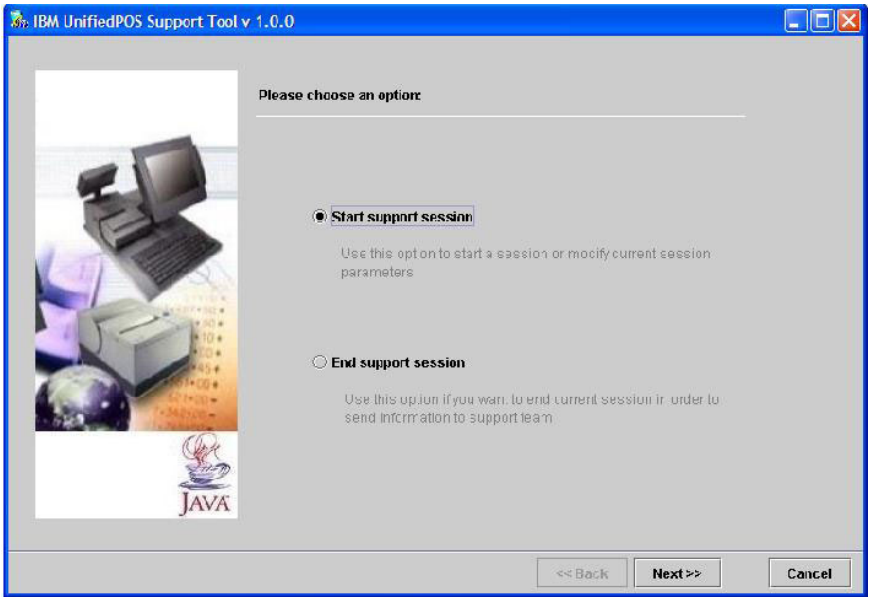


Figure 79.

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:

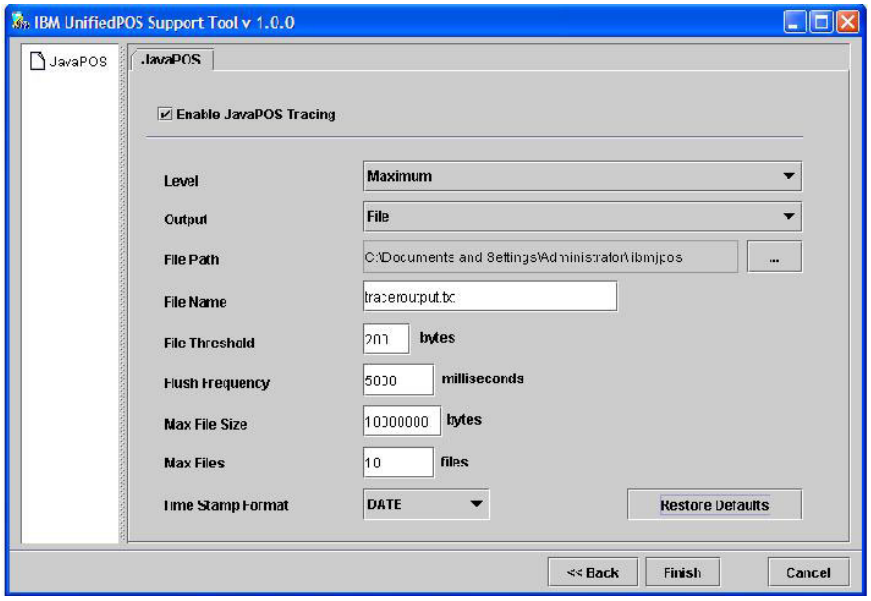


Figure 80.

Enable JavaPOS Tracing defaults to On, even if it is currently Off in the trace file. If you want disable tracing, uncheck this field. *Restore Defaults* loads all the default factory values in one step. Press **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.

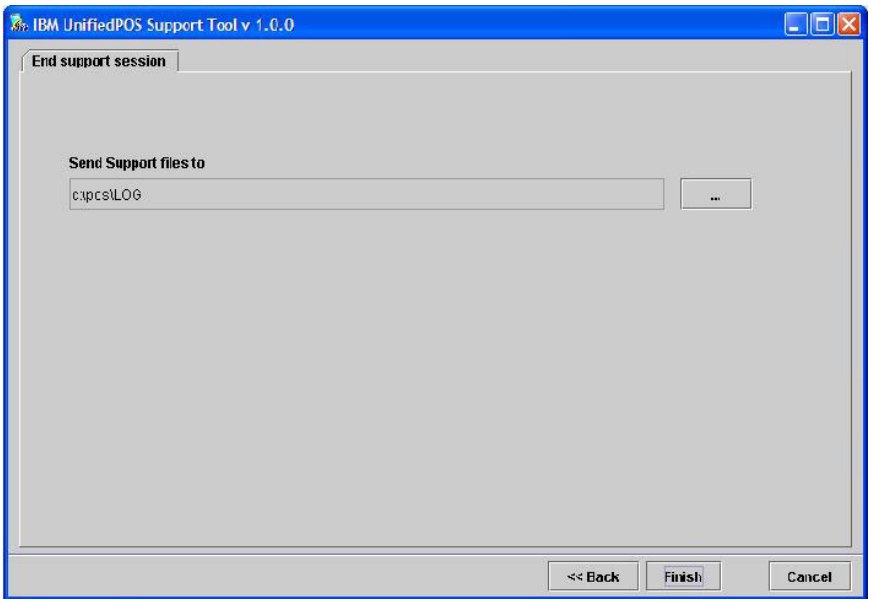


Figure 81.

When you press the <...> (explore button) the following dialog is presented, and only directories can be selected:

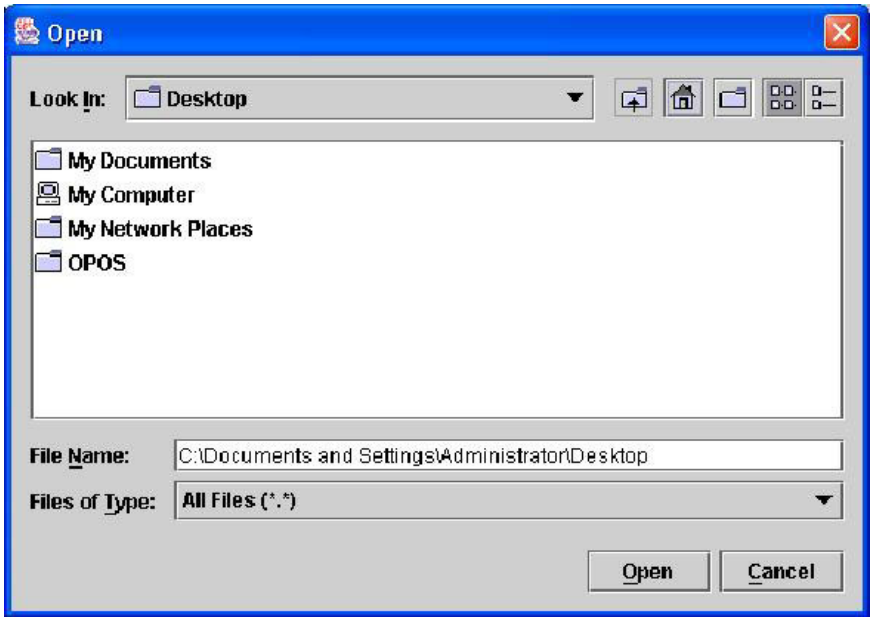


Figure 82. Environment variables file name

After pressing **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 a text file called `myEnvFile.txt` with all the information relative to Environment Variables, jpos.xml, Libraries, and tracer output.

Appendix C. 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

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. GC30-4078-07

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
GC30-4078-07

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

GC30-4078-07

Cut or Fold
Along Line

last update: August 29, 2006
Printed in USA

GC30-4078-07