Honeywell

JavaPOS™ Driver

For Linux® (Honeywell and Legacy Metrologic Products)

User's Guide

Disclaimer

Honeywell International Inc. (“HII”) reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult HII to determine whether any such changes have been made. The information in this publication does not represent a commitment on the part of HII.

HII shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Web Address: www.honeywellaidc.com

Table of Contents

Chapter 6 - Customer Support

1 Introduction

Overview

The Honeywell JavaPOS Driver for Linux is an application-level driver that allows Java-based applications to communicate with Honeywell Scanning & Mobility devices. Any UPOS-compliant application can use the Honeywell JPOS driver to communicate with Honeywell scanners without requiring application code changes.

The Honeywell JavaPOS Driver for Linux supports JavaPOS Spec 1.11.

Note: The Honeywell JavaPOS Driver for Linux does not support firmware flash, configuration, upload or download.

Honeywell JavaPOS for Linux Download Package

The Honeywell JavaPOS for Linux download package file structure contains:

1. Honeywell Linux code to be used with Honeywell RS232 and USB Serial devices.
2. Legacy Metrologic Linux code to be used with Metrologic RS232 and USB Serial devices.

Segments of this document that are specifically for Honeywell products are identified as **Honeywell**.

Segments of this document that are specifically for legacy Metrologic products are identified as **Metrologic**.

Segments with no markings can be used for Honeywell and legacy Metrologic products.

Release Notes

Refer to the Release Notes (*Honeywell JavaPOS Release Notes.txt)*, included in the Honeywell JavaPOS for Linux download package, for the following information:

- Supported Linux versions.
- Supported devices.
- Supported interfaces.
- Known issues and limitations.

2 Honeywell JavaPOS Driver Installation

System Requirements

- Supported Linux operating system (see Release Notes on page 1-1)
- Honeywell JavaPOS 1.7, 1.8, 1.9, 1.10, or 1.11
- Java™ Runtime Environment (JRE) 1.5 or 1.6.

The JREs can be downloaded from http://java.sun.com. Install the JRE by following the instructions in the Java Installation Utility included with the JRE.

Note: To use the Validation Utility shipped with the driver, you must install JRE 1.6.

Installing the Honeywell Service Object and JavaPOS Files

To install the Honeywell JavaPOS driver, you need to have root privileges on your system or enough user rights for shell execution.

1. After the Java Runtime Environment (JRE) is installed, set the Path variable. For example:

```
#edit /etc/profile.d/java.sh
```

2. Put the following two lines in the java.sh file you opened in Step 1.

```
PATH=/usr/java/jre/bin:$PATH
export PATH
```

*Note: Set **usr/java/jre/bin** to the directory where Java is installed.*

3. Download the desired version of the Honeywell JavaPOS driver (***HoneywellJavaPOS[Version]***) from www.honeywellaidc.com and extract the .zip file. The .zip file contains all of the necessary files needed for the JavaPOS driver as well as a test utility.

4. Make sure the downloaded folder has the correct read/execute permissions required by .sh files. The following shows how to add read and execute permissions for the Honeywell JavaPOS driver folder:

```
#chmod -R 777 [Honeywell JavaPOS Driver Folder]
```

5. Provide read/write/execute permission to the RS232 port in use (**/dev/ttyS0** or **/dev/ttyS1**...) and the USB Serial Port in use (**/dev/ttyACM0** or **/dev/ttyACM1**...). Assign execute permission to a non-administrator user for executing the shell script. The following shows how to add permissions to the Linux port:

```
#chmod 777 /dev/ttyS0
```

*Note: **Honeywell Only**. For USB serial interface, a symbolic link is created between /**dev/ACM0** and **/dev/ttyS10** when the OPEN method in the service object is called. When the CLOSE method is called, the symbolic link created in the OPEN method called earlier will be unlinked. Therefore, to use the JavaPOS driver, you must have permission to make the symbolic link..*

6. Install the RXTX library in Linux. Add **librxtxParallel.so** and **librxtxSerial.so** to the Linux library path. The RXTX library is located in the **HoneywellJavaPOS[*Version*]\Component\Linux** folder.

```
# export LD_LIBRARY_PATH=/where/lib/path:$LD_LIBRARY_PATH

# sudo ldconfig
```

7. Copy **honeywelljpos.jar** to your application's folder.

Configuring the Scanner

Configure the scanner before running the JavaPOS test utility to evaluate installation.

Configure a Honeywell Scanner

Verify that the scanner is configured for an RS232 or USB Serial interface by scanning one of the following bar codes.

Scan **TERMID0** to change the interface to **RS232**.

Scan **TERMID130** to change the interface to **USB Serial**.

Configure a Legacy Metrologic Scanner

Configuring an RS232 Scanner using Bar Codes

Scan the following bar codes, in the sequence given, to configure an RS232 scanner.

1. Scan the **Recall Defaults** bar code with an RS232 scanner.

Recall Defaults

2. Scan the **Enable RS232** bar code.

Enable RS232

3. Scan the **Enable STX Prefix** bar code.

Enable STX Prefix

4. Scan the **Enable ETX Suffix** bar code.

Enable ETX Suffix

5. Scan the **Enable Nixdorf ID** bar code.

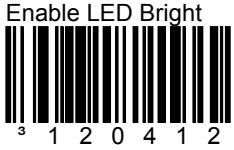
Enable Nixdorf ID

6. Scan the **Enable D/E** bar code.

Enable D/E

7. Scan the **Enable LED bright** bar code.

Enable LED Bright



Configuring a USB Serial Scanner with Bar Codes

Scan the following bar codes, in the sequence given, to configure a Legacy Metrologic USB scanner.

1. Scan the **POSUSB2 B1** bar code with a USB serial scanner.

POSUSB2 B1

2. Scan the **Enable STX Prefix** bar code.

Enable STX Prefix

3. Scan the **Enable ETX Suffix** bar code.

Enable ETX Suffix

4. Scan the **Enable Nixdorf ID** bar code.

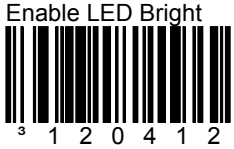
Enable Nixdorf ID

5. Scan the **Enable D/E** bar code.

Enable D/E

6. Scan the **Enable LED bright** bar code.

Enable LED Bright

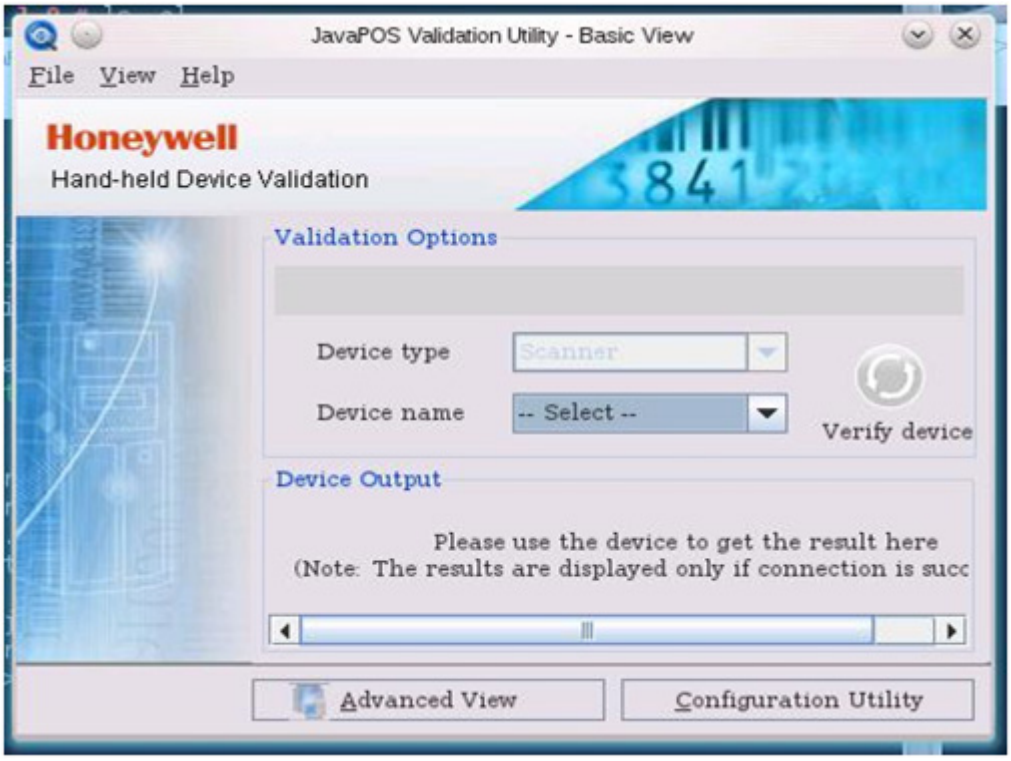


Running the JavaPOS Test Utility to Evaluate Installation

To evaluate JavaPOS driver installation and scanner configuration, follow the steps below.

1. Locate the **TestApp\HoneywellJavaPOSForLinux** folder.
2. Set the path and classpath for your application in the **JPOSUtility.sh**.
3. Run the test application executing **JPOSUtility.sh** from the terminal.
4. Press the **Enter** key on the terminal.

5. The following screen displays.



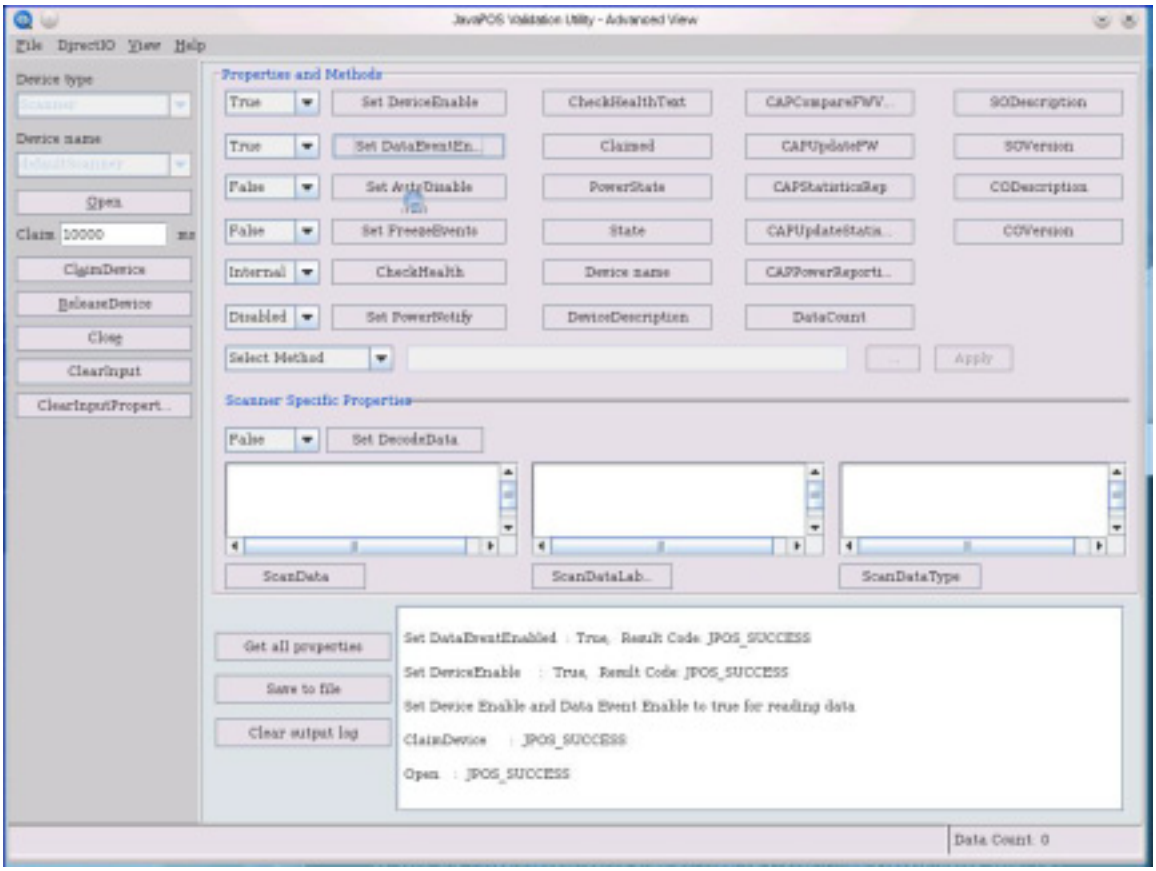
6. Select a device from the **Device name** drop down list.

*Note: You will be presented with a **Test Application Error Message**. It is not a JavaPOS Driver error. Ignore the error message and continue with Step 7. See Test Application Error Message (page 5-5) for the error message text.*

7. Click the **Verify device** button. Results are displayed only if the connection is successful.

8. When the message “Device opened successfully” is displayed, click the **Advanced View** button.

9. The following screen displays.



10. Select a device from the **Device name** drop down list.
11. Click the **Open** button.
12. Click the **ClaimDevice** button.
13. In the Properties and Methods section, select **True** in the drop down list next to **SetDeviceEnable**.
14. In the Properties and Methods section, select **True** in the drop down list next to **Set DataEventEn(able)**.
15. Click **File** and check the **Re-Enable DataEvent** check box.
16. Scan a bar code.

If successful, the bar code is displayed in the text box shown above ScanData.

3

Honeywell/Legacy Metrologic Scanner and the JavaPOS Driver with Your Application

Honeywell Scanner Service Class and JPOS.XML

Bolded fragments in the **JPOS.XML** files in this User's Guide are variables and should be changed to meet your needs before the JPOS.XML file is run.

1. Install the Honeywell JavaPOS driver, following the instructions in Honeywell JavaPOS Driver Installation beginning on page 2-1.
2. Select one service object that corresponds to your JavaPOS control object version from the Class column in the table below. For example, if your application uses JavaPOS control object 1.10, select Class line ending with **HoneywellScanRS232ForLinuxv110** for the JPOS.XML file.

Class	JavaPOS Specification
com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv17	1.7
com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv18	1.8
com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv19	1.9
com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv110	1.10
com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv111	1.11

3. Copy the related JPosEntry option from the jpos.xml to your application's jpos.xml. The jpos.xml file structure begins with the following three lines:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">

<JposEntries>
```

   Each set of device property name variables are enclosed within <JposEntry> and </JposEntry> lines. Using this format allows you to build a jpos.xml file that may contain the property variables for one device or many devices. Refer to the JPOS.XML File Content (For Honeywell RS232 Scanner) Sample (page 3-1) and JPOS.XML File Content (For Honeywell USB Serial Scanner) Sample (page 3-2) that follow.

4. Modify the **jpos.xml** file based on your JavaPOS version, COM ports, baud rates, data bits, parity bits, and stop bits.

JPOS.XML File Content (For Honeywell RS232 Scanner) Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">
<JposEntries>
   <JposEntry logicalName="defaultScanner">
   <creation factoryClass="com.honeywell.javapos.so.HoneywellScannerServiceFactory"
         serviceClass="com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv111"/>
   <vendor name="Honeywell International, Inc."/>
   <jpos category="Scanner" version="1.7"/>
   <product description="Honeywell bar code scanner" name="1900"/>
   <prop name="Interface" type="String" value="RS232"/>
   <prop name="dataBits" type="String" value="8"/>
   <prop name="Model" type="String" value="1900"/>
   <prop name="parity" type="String" value="None"/>
   <prop name="flowControl" type="String" value="None"/>
   <prop name="stopBits" type="String" value="1"/>
   <prop name="deviceBus" type="String" value="RS232"/>
   <prop name="baudRate" type="String" value="115200"/>
   <prop name="Manufacturer" type="String" value="Honeywell"/>
   </JposEntry>
</JposEntries>
```

JPOS.XML File Content (For Honeywell USB Serial Scanner) Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">
<JposEntries>
   <JposEntry logicalName="defaultScanner">
   <creation factoryClass="com.honeywell.javapos.so.HoneywellScannerServiceFactory"
         serviceClass="com.honeywell.javapos.so.HoneywellScanRS232ForLinuxv110"/>
   <vendor name="Honeywell International, Inc."/>
   <jpos category="Scanner" version="1.7"/>
   <product description="Honeywell bar code scanner" name="1900"/>
   <prop name="portName" type="String" value="/dev/ttyS10"/>
   <prop name="USBSerialPort" type="String" value="/dev/ttyACM0"/>
   <prop name="Interface" type="String" value="RS232"/>
   <prop name="dataBits" type="String" value="8"/>
   <prop name="Model" type="String" value="1900"/>
   <prop name="parity" type="String" value="None"/>
   <prop name="flowControl" type="String" value="None"/>
   <prop name="stopBits" type="String" value="1"/>
   <prop name="deviceBus" type="String" value="USB Serial"/>
   <prop name="baudRate" type="String" value="115200"/>
   <prop name="Manufacturer" type="String" value="Honeywell"/>
   </JposEntry>
</JposEntries>
```

Legacy Metrologic Scanner Service Class and JPOS.XML

1. Install the Honeywell JavaPOS driver, following the instructions in Honeywell JavaPOS Driver Installation beginning on page 2-1.
2. Select one service object that corresponds to your JavaPOS control object version from the Class column in the table below. For example, if your application uses JavaPOS control object 1.10, select Class line ending with **MetroScanRS232v110** for the JPOS.XML file.

Class	JavaPOS Specification
com.honeywell.javapos.so.MetroScanRS232v17	1.7
com.honeywell.javapos.so.MetroScanRS232v18	1.8
com.honeywell.javapos.so.MetroScanRS232v19	1.9
com.honeywell.javapos.so.MetroScanRS232v110	1.10
com.honeywell.javapos.so.MetroScanRS232v111	1.11

3. Copy the related JPosEntry option from the jpos.xml to your application's jpos.xml. The jpos.xml file structure begins with the following three lines:

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">

   <JposEntries>
   ```

   Each set of device property name variables are enclosed within <JposEntry> and </JposEntry> lines. Using this format allows you to build a jpos.xml file that may contain the property variables for one device or many devices. Refer to the JPOS.XML File Content (For Legacy Metrologic RS232 Scanner) Sample (page 3-3) and JPOS.XML File Content (For Legacy Metrologic USB Serial Scanner) Sample (page 3-4) that follow.
4. Modify the **jpos.xml** file based on your serial port.

JPOS.XML File Content (For Legacy Metrologic RS232 Scanner) Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">
<JposEntries>
   <JposEntry logicalName="defaultScanner">
   <creation factoryClass="com.honeywell.javapos.so.HoneywellScannerServiceFactory"
         serviceClass="com.honeywell.javapos.so.MetroScanRS232v110"/>
   <vendor name="Honeywell International, Inc."/>
   <jpos category="Scanner" version="1.7"/>
   <product description="Metrologic bar code scanner" name="9540"/>
   <prop name="portName" type="String" value="/dev/ttyS2"/>
   <prop name="Interface" type="String" value="RS232"/>
   <prop name="dataBits" type="String" value="8"/>
   <prop name="Model" type="String" value="9540"/>
   <prop name="HasScale" type="String" value="0"/>
   <prop name="parity" type="String" value="None"/>
   <prop name="flowControl" type="String" value="None"/>
   <prop name="stopBits" type="String" value="1"/>
   <prop name="deviceBus" type="String" value="RS232"/>
   <prop name="baudRate" type="String" value="9600"/>
   <prop name="PortNumber" type="String" value="1"/>
   <prop name="ScannerType" type="String" value="16"/>
   <prop name="ProcessorType" type="String" value="0"/>
   <prop name="FuncMode" type="String" value="4"/>
   <prop name="USBSerial" type="String" value="no"/>
   <prop name="LogFile" type="String" value="JPOS.txt"/>
   <prop name="LOgFileSize" type="String" value="2KB"/>
   <prop name="NumberOfLogFile" type="String" value="2"/>
   <prop name="Manufacturer" type="String" value="Honeywell"/>
   </JposEntry>
</JposEntries>
```

JPOS.XML File Content (For Legacy Metrologic USB Serial Scanner) Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">
<JposEntries>
   <JposEntry logicalName="defaultScanner">
   <creation factoryClass="com.honeywell.javapos.so.HoneywellScannerServiceFactory"
         serviceClass="com.honeywell.javapos.so.MetroScanRS232v110"/>
   <vendor name="Honeywell International, Inc."/>
   <jpos category="Scanner" version="1.7"/>
   <product description="Metrologic bar code scanner" name="9540"/>
   <prop name="portName" type="String" value="/dev/USB0"/>
   <prop name="Interface" type="String" value="RS232"/>
   <prop name="dataBits" type="String" value="8"/>
   <prop name="Model" type="String" value="9540"/>
   <prop name="HasScale" type="String" value="0"/>
   <prop name="parity" type="String" value="None"/>
   <prop name="flowControl" type="String" value="None"/>
   <prop name="stopBits" type="String" value="1"/>
   <prop name="deviceBus" type="String" value="RS232"/>
   <prop name="baudRate" type="String" value="9600"/>
   <prop name="PortNumber" type="String" value="1"/>
   <prop name="ScannerType" type="String" value="16"/>
   <prop name="ProcessorType" type="String" value="0"/>
   <prop name="FuncMode" type="String" value="4"/>
   <prop name="USBSerial" type="String" value="no"/>
   <prop name="LogFile" type="String" value="JPOS.txt"/>
   <prop name="LOgFileSize" type="String" value="2KB"/>
   <prop name="NumberOfLogFile" type="String" value="2"/>
   <prop name="Manufacturer" type="String" value="Honeywell"/>
   </JposEntry>
</JposEntries>
```

4

Legacy Metrologic Scale and the JavaPOS Driver with Your Application

Configuration

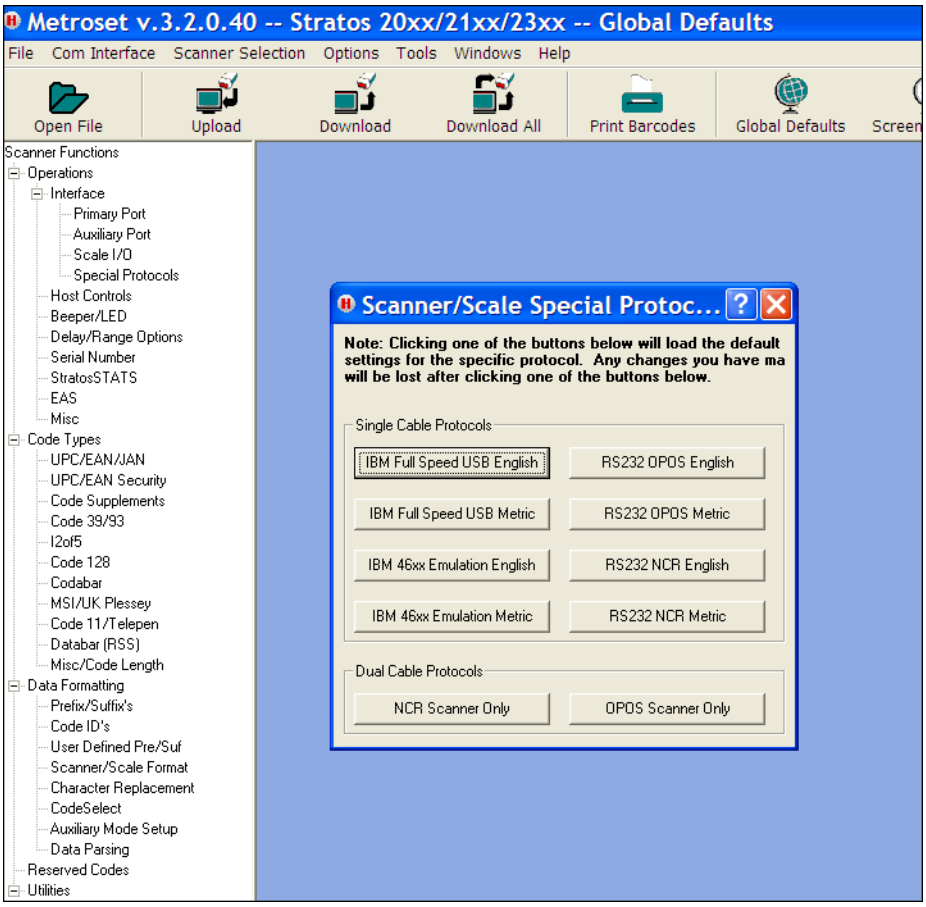
Before the Stratos 2400/2300 scale can use the JavaPOS driver, it needs to be configured to Metric mode or English mode. The modes are configured using the Metroset™ application **or** by scanning the bar codes on the following page.

Using Metroset

Metroset is available for download from Honeywell using the following link: http://www.honeywellaidc.com. Metroset only runs on devices with a Microsoft® Windows operating system.

Click the **RS232 OPOS English** button in the Scanner/Scale Special Protocols screen to set the scale to JPOS English Mode.

Click the **RS232 OPOS Metric** button in the Scanner/Scale Special Protocols screen to set the scale to JPOS Metric Mode.



Using Bar Codes

Scan the JPOS English **or** the JPOS Metric bar code. Scan only one bar code.

Scanning the following bar code configures a single-cable StratosScanner/Scale for Metrologic OPOS/JPOS Single Cable RS232 Mode – **English**.

JPOS English

Scanning the following bar code configures a single-cable Stratos Scanner/Scale for Metrologic OPOS/JPOS Single Cable RS232 Mode – **Metric**.

JPOS Metric

*Note: After scanning one of the bar codes above for a Stratos Scanner/Scale, one of the following lines should be added to the **Sample JPOS.XML File Content (For Scale)** that follows.*

To use the Stratos Scale in English Mode, set UsesKilograms string value to **0**.

```
<prop name="UsesKilograms" type="String" value="0"/>
```

To use the Stratos Scale in Metric Mode, set UsesKilograms string value to **1**.

```
<prop name="UsesKilograms" type="String" value="1"/>
```

Using the JavaPOS Driver with Your Application

1. Install the Honeywell JavaPOS driver, following the instructions in Honeywell JavaPOS Driver Installation beginning on page 2-1. Select the service object **com.honeywell.javapos.so.MetroScale17** in the Class column below.

Class	JavaPOS Specification
com.honeywell.javapos.so.MetroScale17	1.7

2. Copy the related JPosEntry option from the jpos.xml to your application's jpos.xml. The jpos.xml file structure begins with the following three lines:

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">

   <JposEntries>
   ```

   Each set of device property name variables are enclosed within <JposEntry> and </JposEntry> lines. Using this format allows you to build a jpos.xml file that may contain the property variables for one device or many devices. Refer to the JPOS.XML File Content (For Scale) Sample (page 4-3) that follows.

3. Modify the **jpos.xml** file based on your JavaPOS version, scanner model, and COM Port..

JPOS.XML File Content (For Scale) Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN" "jpos/res/jcl.dtd">
<JposEntries>
   <JposEntry logicalName="defaultScale">
   <creation factoryClass="com.honeywell.javapos.so.HoneywellScaleServiceFactory"
         serviceClass="com.honeywell.javapos.so.MetroScale17"/>
   <vendor name="Honeywell International, Inc."/>
   <jpos category="Scale" version="1.7"/>
   <product description="Metrologic bar code scanner/scale" name="MS2020"/>
   <prop name="portName" type="String" value="/dev/ttyS2"/>
   <prop name="dataBits" type="String" value="8"/>
   <prop name="parity" type="String" value="None"/>
   <prop name="flowControl" type="String" value="None"/>
   <prop name="HasScaleDisplay" type="String" value="Yes"/>
   <prop name="stopBits" type="String" value="1"/>
   <prop name="deviceBus" type="String" value="RS232"/>
   <prop name="UsesKilograms" type="String" value="0"/>
   <prop name="baudRate" type="String" value="9600"/>
   <prop name="LogFile" type="String" value="JPOS.txt"/>
   <prop name="LogFileSize" type="String" value="5KB"/>
   <prop name="NumberOfLogFile" type="String" value="3"/>
   </JposEntry>
</JposEntries>
```

5 Miscellaneous

Folders

Name	Description
jpos	Folder containing jpos.properties, which is used for serviceManager class.
resources	Folder containing four properties files, which are used by JavaPOSSuite.jar.

Driver Files

Name	Description
honeywelljpos.jar	Honeywell JavaPOS Service Object file.
JavaPOSSuite.jar	Test Application file.
jpos.xml	The JavaPOS configuration file.
jpos110.jar	JavaPOS Driver Control Object file.
JPOSUtility.sh	Used to run the test application.
librxtxParallel.so	Used to communicate with Parallel ports.
librxtxSerial.so	Used to communicate with COM ports.
log4j-1.2.15.jar	Used by JavaPOSSuite.jar for logging.
MetroJPOSTest.sh	Used to run Test App when checking for BELL command.
RXTXcomm.jar	Used to communicate with COM ports.
xerces.jar	XML parser file.

Supported Methods and Properties

Methods

Methods common to Honeywell Scanners and Metrologic Scanners are listed below:

Method	Honeywell Scanner	Metrologic Scanner
checkHealth	X	X
claim	X	X
clearInput	X	X
clearInputProperties	X	X
clearOutput	X	X
close	X	X
directIO	X	X
open	X	X
release	X	X
retrieveStatistics	X	X

Properties

Properties common to Honeywell Scanners and Metrologic Scanners are listed below:

Property	Honeywell Scanner	Metrologic Scanner
CapPowerReporting	X	
CapStatisticsReporting	X	X
CheckHealthText	X	X
Claimed	X	X
DataCount	X	X
DataEventEnabled	X	X
DecodeData	X	X
DeviceControlDescription	X	X
DeviceControlVersion	X	X
DeviceEnabled	X	X
DeviceServiceDescription	X	X
DeviceServiceVersion	X	X
FreezeEvents	X	X
PhysicalDeviceDescription	X	X
PhysicalDeviceName	X	X
Power Notify	X	
PowerState	X	X
ScanData	X	X
ScanDataLabel	X	X
ScanDataType	X	X
State	X	X

Enable Logging

Use the LogFile tag to enable logging of data received from the scanner. If the LogFile entry is not present in **jpos.xml** or its value is an empty string, the JPOS Driver does not log any data received from the scanner.

Example 1

```
<prop name="LogFile" type="String" value="C:\LogFolder\JPOS.txt"/>
```

The above setting instructs the JPOS driver to send logging data to the JPOS.txt file located in the C:\LogFolder\ folder. If the LogFolder folder does not exist in the /root drive, it is created when called.

Example 2

```
<prop name="LogFile" type="String" value="JPOS.txt"/>
```

When a file name is specified as the property value but does not yet exist, the file is created in the current folder when called.

Example 3

```
<prop name="LogFileSize" type="String" value="10KB"/>
```

This entry creates a 10KB log file. MB and GB are also supported values. The default log file size is 10MB.

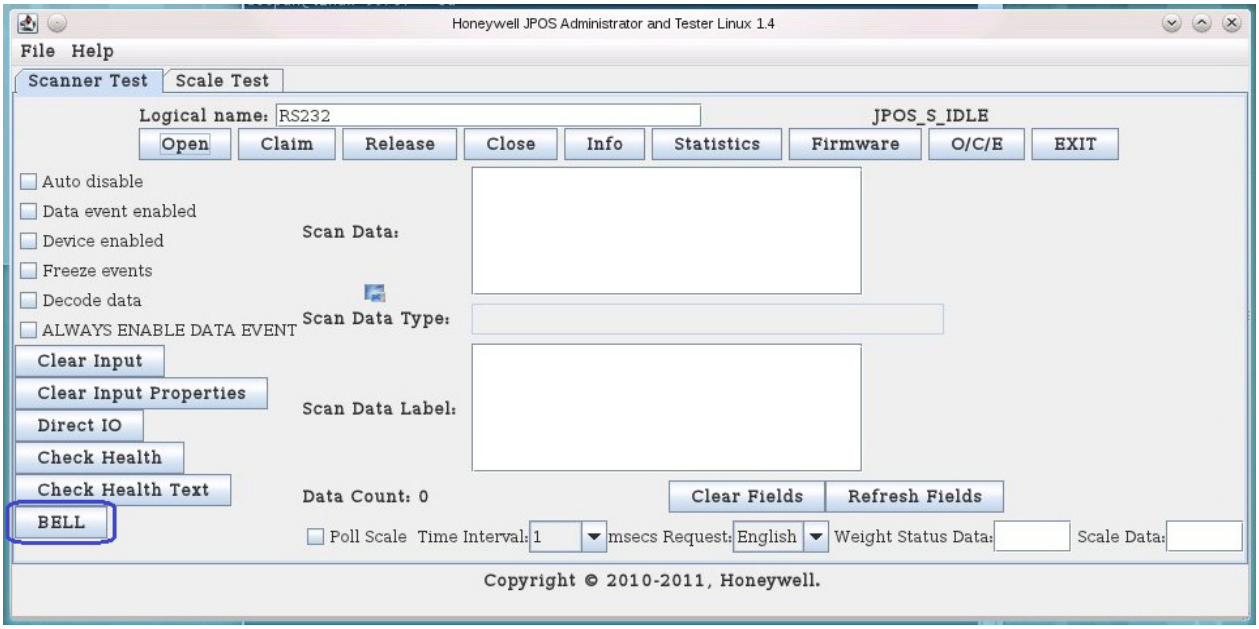
Example 4

```
<prop name="NumberOfLogFile" type="String" value="3"/>
```

In this example three log files are created; two of which will be backup files.

DirectIO BEEP Command

1. Run the JPOS test application by executing **MetroJPOStest.sh.** The following screen displays.



2. Enter the scanner's **Logical name** (RS232/USB Serial) as listed in **jpos.xml**.
3. Click **Open**.
4. Click **Claim**.
5. Click **Enable Device**.
6. If the device is successfully claimed and enabled, click **BELL**. The scanner's bell will sound.

Example Code for Using the Bell Command in an Application

```
try
{
String ret[] = new String[1];
Scanner.DirectIO(30,null,ret);
}
catch(Exception e){}
```

Disable Suffix and Prefix

1. Scan the **Disable ETX Suffix** bar code.

Disable ETX Prefix

2. Scan the **Disable STX Prefix** bar code.

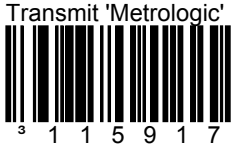
Disable STX Prefix

Legacy Metrologic - Connect/Disconnect RS232 Scanner

The JPOS driver can be configured to check whether a scanner is connected on the RS232 interface.

1. Scan the following bar code to enable **Transmit "Metrologic" on "l"**.

Transmit 'Metrologic'



2. Add the following line to the **jpos.xml** file.

```
<prop name="ConnectionDetectRS232" type="String" value="true"/>
```

Legacy Metrologic – Check Health / BEL Command

Configure the JPOS driver to check health and enable the BEL command. Follow the steps below.

1. **Transmit "Metrologic" on "l"** must be enabled to support the "l" command. Scan the following bar code.

Transmit 'Metrologic'

2. Scan the following bar code to enable **Beep on BEL**.

Beep on BEL

Test Application Error Message

6 Customer Support

Technical Assistance

If you need assistance installing or troubleshooting your device, please contact us by using one of the methods below:

Knowledge Base: www.hsmknowledgebase.com

Our Knowledge Base provides thousands of immediate solutions. If the Knowledge Base cannot help, our Technical Support Portal (see below) provides an easy way to report your problem or ask your question.

Technical Support Portal: www.hsmsupportportal.com

The Technical Support Portal not only allows you to report your problem, but it also provides immediate solutions to your technical issues by searching our Knowledge Base. With the Portal, you can submit and track your questions on-line and send and receive attachments.

Web form: www.hsmcontactsupport.com

You can contact our technical support team directly by filling out our on-line support form. Enter your contact details and the description of the question/problem.

Telephone: www.honeywellaidc.com/locations

For our latest contact information, please check our website at the link above.

End User License Agreement

This License Agreement (“Agreement”) is a legal agreement between you and Honeywell International Inc. (“Licensor”) regarding the associated software (“Software”), which may include software owned by Honeywell and software licensed by Honeywell from its software suppliers (“Suppliers”).

The Agreement also applies to any updates, supplements, Internet-based services, and support services for this software, unless other terms accompany those items. If so, those terms apply.

By installing, activating, or using the Software, you agree to be bound by the terms and conditions of this Agreement. If you do not agree to be so bound, you may not install the Software or, if the Software is already installed, you must promptly remove it. The Software and any accompanying materials (including, without limitation, any images, photographs, animations, video, audio, music, text, and applets incorporated into the Software, the accompanying media, and printed materials) are owned by Licensor and its Suppliers and protected under U.S. and international copyright laws, and may be protected under additional intellectual property laws. The Software is licensed, not sold, and Licensor and its Suppliers retain all right, title, and interest therein other than those rights specifically granted to you under this Agreement. You accept responsibility for selection of the Software to achieve your intended results, and for installation, activation, use of, and results obtained from, the Software.

LICENSE: Licensor hereby grants you a non-exclusive License to use this Software, without right of sub-license, only in object or executable code form, and only in or with Licensor's products (“Products”). You may not rent, lease, or lend the Software. You may permanently transfer rights under this Agreement only as part of a permanent sale or transfer of the Products, and only if the recipient accepts this Agreement. If the Software is an upgrade, any transfer must also include all prior versions of the Software.

You agree that the Software and any Software-related materials provided under this Agreement are and shall at all times remain the sole and exclusive property of Licensor and its Suppliers. Unauthorized copying of the Software is expressly forbidden. The Software may be patent-pending and/or patented; please refer to documentation accompanying the product, including labels and user guides, for specifics. You may be held legally responsible for any infringement of copyright or other intellectual property rights caused by your failure to abide by this Agreement.

If the Software is identified by Licensor as a demonstration version, Licensee may use the Software on multiple products or platforms. If the Software is provided by Licensor as other than a demonstration version, Licensee may use the Software only on a single Licensor product. User's guides and programming guides for the Software that are provided by Licensor in either hard or electronic copy may be copied and distributed.

RESTRICTIONS: You shall not use, print, copy, or display the Software in whole or in part except as expressly permitted in writing. You shall not modify, translate, alter, create derivatives of, “reverse compile,” decompile, merge with another program, or otherwise derive the source code for the Software, or defeat any “keys” or codes controlling authorized access or functionality, nor will you allow others to do the same.

NOTE ON JAVA SUPPORT: The Software may contain support for programs written in Java. Java technology is not fault tolerant and is not designed, manufactured, or intended for use or resale as online control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of Java technology could lead to directly to death, personal injury, or severe physical or environmental damage.

DISCLAIMERS AND LIMITATIONS OF LIABILITY: THE SOFTWARE IS NOT FAULT TOLERANT, AND IS PROVIDED AS IS AND WITH ALL FAULTS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, AND EXCEPT AS OTHERWISE EXPRESSLY SET FORTH IN THIS AGREEMENT, LICENSOR AND ITS SUPPLIERS (1) DISCLAIM ANY AND ALL PROMISES AND REPRESENTATIONS WITH RESPECT TO THE SOFTWARE, INCLUDING ITS CONDITION, ITS CONFORMITY TO ANY REPRESENTATION OR DESCRIPTION, AND THE EXISTENCE OF ANY LATENT OR PATENT DEFECTS, (2) DISCLAIM ALL WARRANTIES, WRITTEN OR ORAL, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT OF THIRD-PARTY RIGHTS; AND, (3) SHALL HAVE NO LIABILITY FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE SOFTWARE. THIS LIMITATION SHALL APPLY EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. IN NO EVENT SHALL LICENSOR OR ITS SUPPLIERS BE LIABLE FOR ANY AMOUNT IN EXCESS OF (1) THE INITIAL LICENSE FEE THAT LICENSOR RECEIVED FROM YOU FOR THE PRODUCTS, IN THE CASE OF LICENSOR, OR (2) U.S. TWO HUNDRED AND FIFTY DOLLARS (U.S. $250.00), IN THE CASE OF LICENSOR'S SUPPLIERS. ALL RISK AS TO QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU.

GENERAL: This Agreement is the complete agreement and understanding of the parties with respect to the Software and supersedes all prior oral, written, or other representations and agreements. You acknowledge that the Software is of U.S. origin, and agree to comply with all applicable international and national laws that apply to the Software, including the U.S. Export Administration Regulations, as well as end-user, end-use, and country destination restrictions issued by the U.S. and other governments. If this product is acquired under the terms of a U.S. Government contract, use, duplication, and disclosure are subject to the terms of this license and the restrictions contained in the Rights in Technical Data and Computer Software clause at 252.227-7013 (DOD contracts); and subdivisions (a) through (d) of 52.227-19 as applicable. This Agreement shall be governed by the laws of the State of New York, without regard to its conflicts of law provisions.

DISTRIBUTORS AND RESELLERS: In addition to the License rights granted in this License Agreement, Distributors and Resellers of Licensor's Products shall have the right to install and sublicense the Software to End Users solely for the purpose of using the Software on Licensor's products for the End User's own business. Distributors and Resellers of Licensor's Products shall have the right to advertise or otherwise market the Software for use on Licensor's products. User's guides and programming guides for the Software that are provided by Licensor in either hard or electronic copy may be copied and distributed. Distributors and Resellers of Licensor's Products shall provide this License Agreement with each sub-license to an End User of the Software. Distributors and Resellers of Licensor's Products shall promptly discontinue distribution of the Software to any End User which does not comply with the obligations in this License Agreement and shall notify Licensor and cooperate with Licensor in investigating instances of violation thereof.

Honeywell Scanning & Mobility
9680 Old Bailes Road
Fort Mill, SC 29707

www.honeywellaidc.com