Datalogic™ JavaPOS Device Services

User Manual

Datalogic USA, Inc.
959 Terry Street
Eugene, OR 97402
USA
Telephone: (541) 683-5700
Fax: (541) 345-7140

©2006 - 2018 Datalogic S.p.A. and/or its affiliates

An Unpublished Work - All rights reserved. No part of the contents of this documentation or the procedures described therein may be reproduced or transmitted in any form or by any means without prior written permission of Datalogic S.p.A. and/or its affiliates ("Datalogic" or "Datalogic USA, Inc.”). Owners of Datalogic products are hereby granted a non-exclusive, revocable license to reproduce and transmit this documentation for the purchaser's own internal business purposes. Purchaser shall not remove or alter any proprietary notices, including copyright notices, contained in this documentation and shall ensure that all notices appear on any reproductions of the documentation.
Should future revisions of this manual be published, you can acquire printed versions by contacting your Datalogic representative. Electronic versions may either be downloadable from the Datalogic website (www.datalogic.com) or provided on appropriate media. If you visit our website and would like to make comments or suggestions about this or other Datalogic publications, please let us know via the "Contact Datalogic" page.

Disclaimer

Datalogic has taken reasonable measures to provide information in this manual that is complete and accurate, however, Datalogic reserves the right to change any specification at any time without prior notice.
Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S. and the E.U. All other brand and product names may be trademarks of their respective owners.
Magellan is a trademark of Datalogic S.p.A. or of Datalogic and/or its affiliates, registered in many countries, including the U.S. and the E.U.

Table of Contents

Chapter 1
Datalogic™ JavaPOS Device Services

Introduction

Document Conventions

Formatting conventions are used throughout this guide to provide a consistent method for representing screen shots and command entries.

NOTE

Notes contain additional information of interest to the reader.

Filenames, paths, field selections, and data or keystrokes entered by the user are shown in this `monospaced` typeface.

Controls including command bar sequences, prompts, fields, checkboxes and radio-buttons are printed in this **bold** typeface.

About Datalogic JavaPOS Device Services

These Device Services are current to UPOS version 1.14. These Device Services provide support for all functionality described in the *Unified POS Retail Peripheral Architecture*, version 1.14 and the JavaPOS appendix to that spec. It is available online from the National Retail Federation at: http://www.javapos.com/.

In addition, Direct IO support is provided for all commands documented in IBM's Universal Serial Bus OEM Point-of-Sale Device Interface Specification, Version 1.29., and for limited commands in the RS-232 and RS-232 Single Cable interfaces. Instructions for accessing Direct IO commands are provided later in this document.

Datalogic Products Supported

Most Datalogic products with RS-232 STD, RS-232 SC, and OEM USB interfaces are supported. Products released after the publication of this manual may still be supported by JavaPOS. Please contact Datalogic Technical support or your Datalogic representative for questions about specific models.

Limitations

The following limitations apply to this version of the Datalogic JavaPOS Device Services:

- For an RS-232 Standard scale device, only the SASI scale interface is supported.
- For the QS2500 and Gryphon, the only supported interface is OEM/IBM USB (no serial interface).
- Windows operating system support is limited to Windows XP and later versions.

Installation

Introduction

Datalogic JavaPOS is a set of Java based drivers that allow Java application developers to communicate with Datalogic-based Point of Sale (POS) Scales and Scanners. As long as the Java application adheres to the JavaPOS specification up to 1.14, it does not need to be written with the Datalogic drivers in mind, but rather should work with any JavaPOS 1.14 or less compliant drivers. The Datalogic JavaPOS drivers can be installed on any system with a pre-existing Java-based Point of Sale application, and (with some limited configuration) the Point of Sale application should have no problems interacting with Datalogic scales and scanners.

These drivers were written, tested and verified to work on the following operating systems:

- Microsoft Windows XP SP3, Vista, Windows 7, Windows 10
- Fedora Core Linux versions 5 through 9
- SuSE Linux Professional 9.3, CentOS 6, OpenSuSE 11, Ubuntu

For a specific list of Datalogic scale and scanner brands known to work under Windows and Linux, please contact Datalogic Technical Support or your representative.

This document discusses three topics related to installation of the Datalogic JavaPOS drivers:

1. "Installation" of the software under both Windows and Linux.
2. Quick demonstration of a test application in "Verification" on page 9.
3. "Technical Details" on page 13 — a detailed description of what happens during the installation process.

Windows Installation

Datalogic JavaPOS has the following requirements for installation on a Windows system:

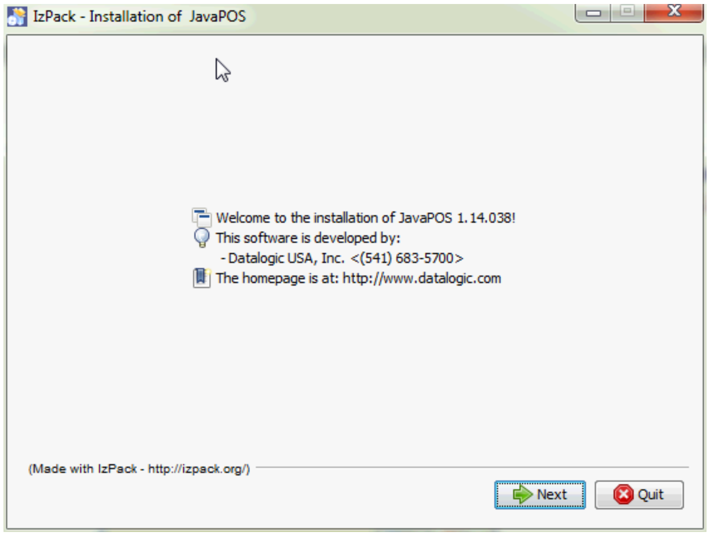
- Sun Java runtime environment version 1.5 or later (available from http://www.oracle.com/technetwork/java/index.html)
- Administrator privileges for the current user

NOTE **If Datalogic JavaPOS is already installed on your computer, you must uninstall it before installing again.**

To install Datalogic JavaPOS, perform the following steps:

1. Double-click the `setup.jar` file. The installer starts and a **Welcome** dialog is displayed.

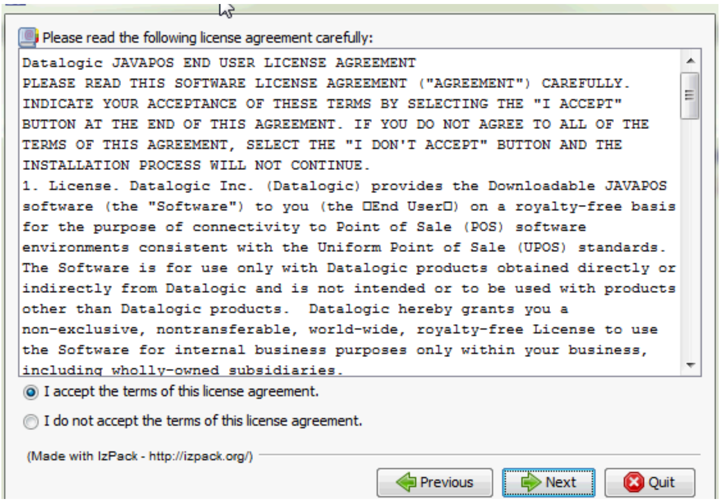
NOTE **On systems running Windows 7 and later, it may be necessary to install JavaPOS as a user with Administrator privileges.**



Welcome to the installation of JavaPOS 1.14.038!
This software is developed by:
- Datalogic USA, Inc. <(541) 683-5700>
The homepage is at: http://www.datalogic.com

(Made with IzPack - http://izpack.org/)

2. Click **Next**. You must click to indicate you accept the terms of the End User License Agreement (EULA) in order to continue with the installation.



Please read the following license agreement carefully:

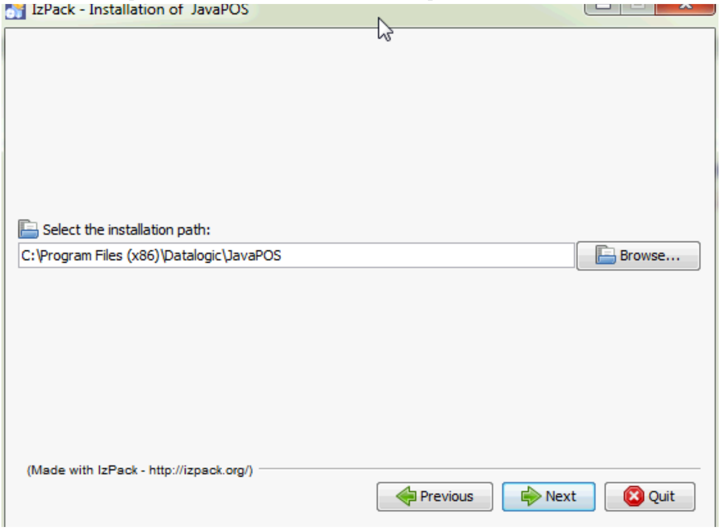
```
Datalogic JAVAPOS END USER LICENSE AGREEMENT
PLEASE READ THIS SOFTWARE LICENSE AGREEMENT ("AGREEMENT") CAREFULLY.
INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "I ACCEPT"
BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE
TERMS OF THIS AGREEMENT, SELECT THE "I DON'T ACCEPT" BUTTON AND THE
INSTALLATION PROCESS WILL NOT CONTINUE.
1. License. Datalogic Inc. (Datalogic) provides the Downloadable JAVAPOS
software (the "Software") to you (the □End User□) on a royalty-free basis
for the purpose of connectivity to Point of Sale (POS) software
environments consistent with the Uniform Point of Sale (UPOS) standards.
The Software is for use only with Datalogic products obtained directly or
indirectly from Datalogic and is not intended or to be used with products
other than Datalogic products.  Datalogic hereby grants you a
non-exclusive, nontransferable, world-wide, royalty-free License to use
the Software for internal business purposes only within your business,
including wholly-owned subsidiaries.
```

● I accept the terms of this license agreement.

○ I do not accept the terms of this license agreement.

(Made with IzPack - http://izpack.org/)

3. Click **Next**. You are prompted to enter an installation path (the default is `C:\Program Files\Datalogic\JavaPOS`).

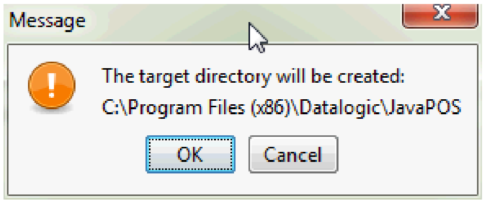


IzPack - Installation of JavaPOS

Select the installation path:

C:\Program Files (x86)\Datalogic\JavaPOS

(Made with IzPack - http://izpack.org/)

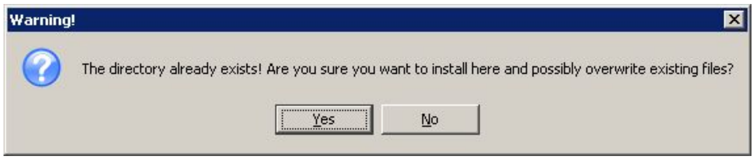


Message

The target directory will be created:
C:\Program Files (x86)\Datalogic\JavaPOS

CAUTION

If Datalogic JavaPOS is already present on your computer, a warning is displayed.

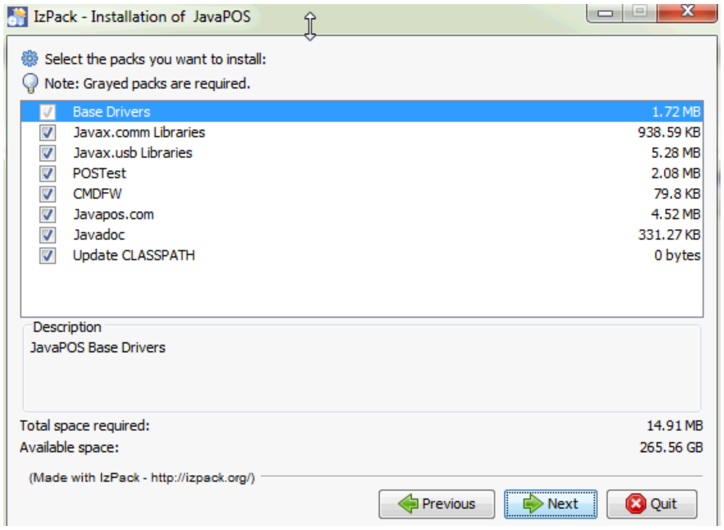


Warning!

The directory already exists! Are you sure you want to install here and possibly overwrite existing files?

If you receive this message, click No to halt the installation. Uninstall any previous version of Datalogic JavaPOS, then return to step 1 of this procedure.

4. Available packages for the current version are displayed. Only the Base Datalogic Drivers are absolutely required. If installing on a system for the first time to test the functionality of the drivers, it is recommended that all packages be installed. It is important to realize that these drivers are dependent on many of these files. If some packages are unchecked, it may render the drivers unusable, unless the required files are already installed on the system.



IzPack - Installation of JavaPOS

Select the packs you want to install:

Note: Grayed packs are required.

☑	Base Drivers	1.72 MB
☑	Javax.comm Libraries	938.59 KB
☑	Javax.usb Libraries	5.28 MB
☑	POSTest	2.08 MB
☑	CMDFW	79.8 KB
☑	Javapos.com	4.52 MB
☑	Javadoc	331.27 KB
☑	Update CLASSPATH	0 bytes

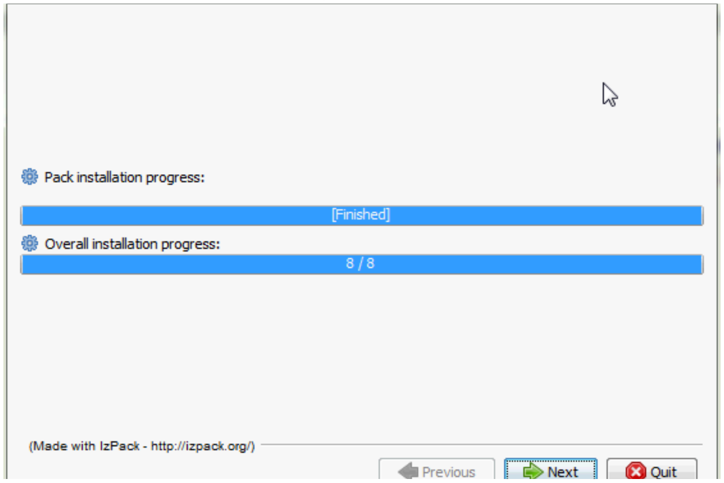
Description

JavaPOS Base Drivers

Total space required:	14.91 MB
Available space:	265.56 GB

(Made with IzPack - http://izpack.org/)

5. Click **Next**. An installation progress dialog is displayed.



Pack installation progress:

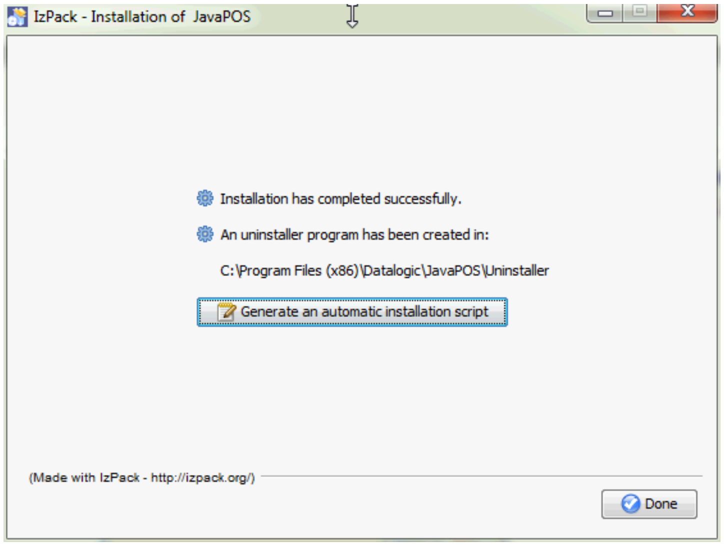
[Finished]

Overall installation progress:

8 / 8

(Made with IzPack - http://izpack.org/)

6. When the **Installation Completed** message is displayed, click **Done**.



IzPack - Installation of JavaPOS

Installation has completed successfully.

An uninstaller program has been created in:

C:\Program Files (x86)\Datalogic\JavaPOS\Uninstaller

Generate an automatic installation script

(Made with IzPack - http://izpack.org/)

Linux Installation

As with the Windows operating system, Linux requires version 1.5 or later of the Java Runtime Environment. Visit http://www.oracle.com/technetwork/java/index.html for download and installation instructions. The Java application must exist in your system path in order for the Datalogic JavaPOS application links to function properly.

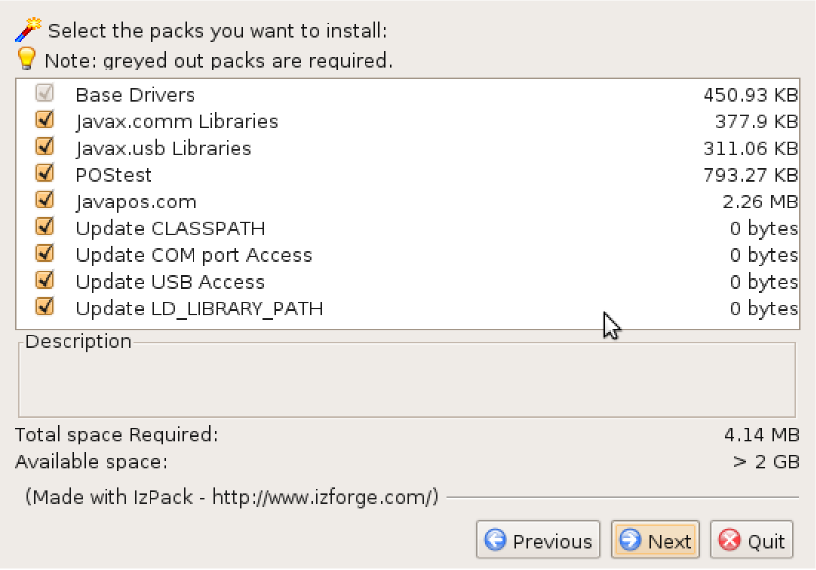
If Fedora Core 3, CentOS or Ubuntu is installed as a Workstation, it installs a GNU java runtime environment. This JRE is not adequate for installing or running Datalogic JavaPOS. Install the SUN JRE. After this is complete, it will be necessary to ensure that the SUN version of Java is in the system execution path ahead of the one in `/usr/bin`. It may even be desirable to delete the "`/usr/bin/java`" file prior to installing the SUN JRE, then ensuring the SUN Java executable is in the system execution path.

Under the Linux operating system, the installation procedure is exactly the same as for Windows. If the installer is not logged on as the root user, the installation procedure terminates with a message reminding you to log on as root.

From a terminal window under either Gnome or KDE, execute the following command from the same folder where the `setup.jar` file exists (remember, Java must be found in the system path):

```
java -jar setup.jar
```

This command starts the automated Java-based installer. The screens and prompts are the same as for windows, with the exception of the "packages" window. It resembles the following window:



Select the packs you want to install:

Note: greyed out packs are required.

☑	Base Drivers	450.93 KB
☑	Javax.comm Libraries	377.9 KB
☑	Javax.usb Libraries	311.06 KB
☑	POStest	793.27 KB
☑	Javapos.com	2.26 MB
☑	Update CLASSPATH	0 bytes
☑	Update COM port Access	0 bytes
☑	Update USB Access	0 bytes
☑	Update LD_LIBRARY_PATH	0 bytes

Description

Total space Required: 4.14 MB

Available space: > 2 GB

(Made with IzPack - http://www.izforge.com/)

After running the installer, it is recommended that you log off and log back on to ensure the settings take effect.

Verification

The Datalogic JavaPOS installer automatically installs a POSTest application for testing all kinds of JavaPOS devices. If one of the previously discussed Datalogic/Datalogic scanners or scales is available, you can test the drivers to make sure they were installed correctly.

On Windows-based installation, the install directory contains a POSTest.bat file that can be used to launch POSTest.

On Linux based installations, shell scripts in the `/opt/dls/JavaPOS` directory can be used to run POSTest or JEREditor.

You can run POSTest from a command prompt or terminal window by issuing the following command:

```
sudo./POSTest.sh
```

NOTE

POSTest should always be run with sudo privileges. Shell script may also need the executable file permission added to it by running

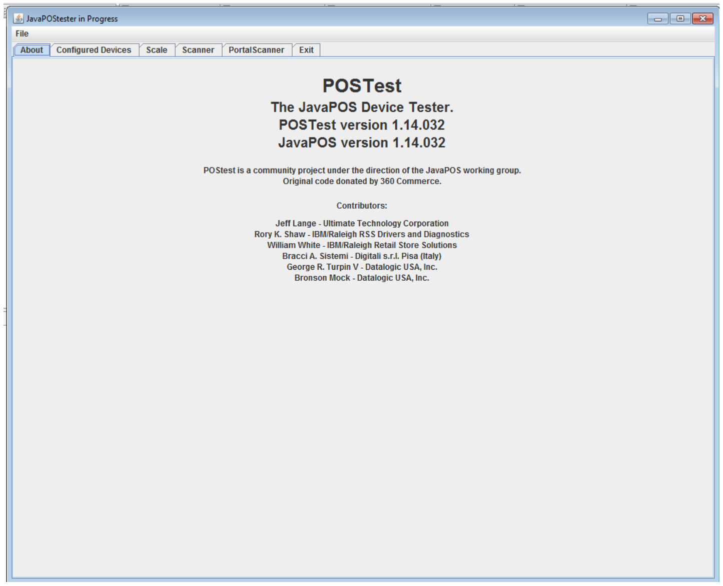
```
sudo chmod +x POSTest.sh
```

`POSTest.sh (.csh)` is located in the JavaPOS directory (depending upon which shell is used).

NOTE

If this command does not launch the application, the `CLASSPATH` is probably not configured properly. Restart your computer and try again.

After POSTest starts, the following screen is displayed:



POSTest

The JavaPOS Device Tester.
POSTest version 1.14.032
JavaPOS version 1.14.032

POStest is a community project under the direction of the JavaPOS working group.
Original code donated by 360 Commerce.

Contributors:

Jeff Lange - Ultimate Technology Corporation
Rory K. Shaw - IBM/Raleigh RSS Drivers and Diagnostics
William White - IBM/Raleigh Retail Store Solutions
Bracci A. Sistemi - Digitali s.r.l. Pisa (Italy)
George R. Turpin V - Datalogic USA, Inc.
Bronson Mock - Datalogic USA, Inc.

NOTE

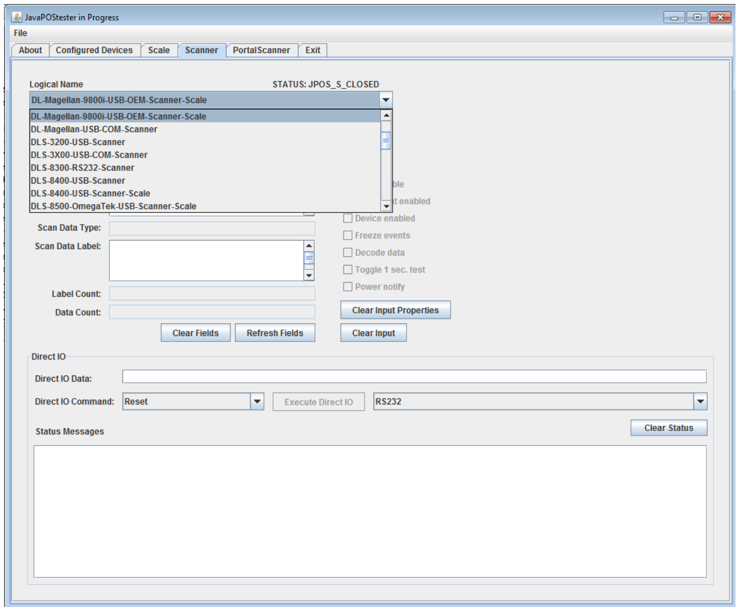
Most of the devices listed by this application are not yet supported. This procedure only addresses the Scale and Scanner devices.

Example - Configuring a Scanner

The following procedure describes how to configure a scanner, using the Datalogic Magellan 9800i Scanner-Scale under Windows as an example. Perform the following steps:

1. Plug the scanner into an active USB port. If Windows prompts you to restart your computer, do so.

2. Click the **Scanner** tab. The Scanner properties are displayed.



Logical Name: DL-Magellan-9800i-USB-OEM-Scanner-Scale

STATUS: JPOS_S_CLOSED

Direct IO Command: Reset

RS232

3. In the **Logical Name** field, select `DL-Magellan-9800i-USB-OEM-Scanner-Scale`.

4. Click **Open**, and then **Claim**.

5. Select the **Data event enabled**, **Device enabled**, and **Decode data** checkboxes, or click the **Enable** button.

6. Scan an item. If the scanner is properly connected and configured, the scan data is automatically displayed.

If a “Failed” message is displayed, POSTest could not find the device’s logical name in the `jpos.xml` file.

To correct this error, locate all `jpos.xml` files on the system and identify the one that is picked up by the application during execution. A file is in the same directory where the application is launched (or listed first in the `CLASSPATH`) takes priority. The installer creates a `jpos.xml` file in the installation folder. This file contains the logical names for many different Datalogic devices. You may have to move some or all of the logical names to the active `jpos.xml` file.

JER Editor

The JER Editor helps you configure the devices listed in the jpos.xml file. You can edit this file manually, but JER Editor provides an easier mechanism for editing the devices' configuration properties. You can use it to add new devices or change COM port assignments.

If under Linux there is no shortcut link, you may have to log back onto the GUI desktop as the root user. The "root" user must create shortcut links for any other user who will need to run the Datalogic JavaPOS applications. The Linux installer only creates menu shortcut links under KDE, but desktop links are created under both KDE and Gnome.

You can also run `POSTest` from a command prompt or terminal window by issuing the following command:

```
java com.Datalogic.jpos.editor.DeviceEditor
```

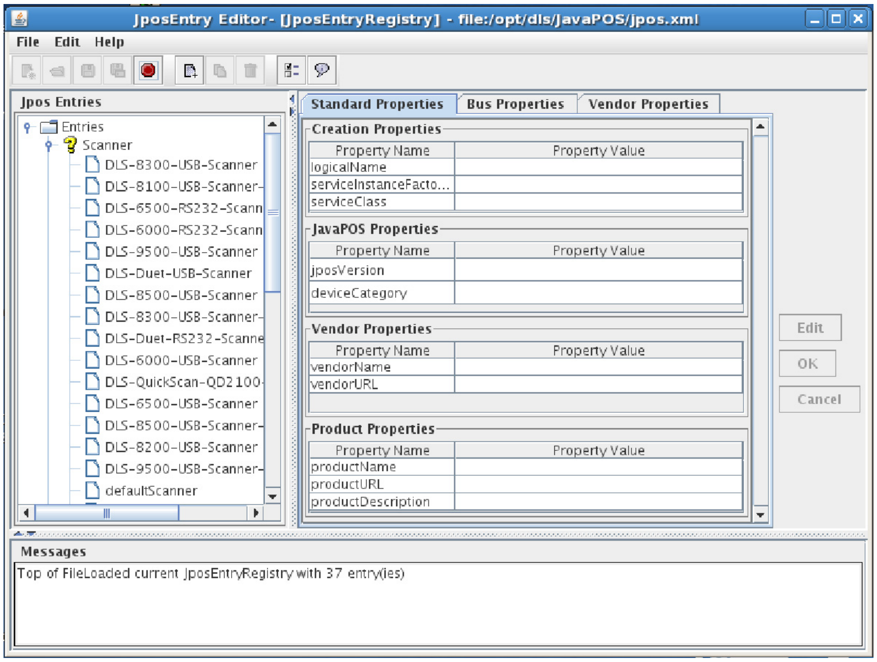
There is a new script that can be used to start JER Editor:

`JEREditor.sh (.csh)` is located in the JavaPOS directory (depending upon which shell is used).

NOTE

If this command does not launch the application, the `CLASSPATH` is probably not configured properly. Restart your computer and try again.

After JER Editor starts, a list of device types is displayed. Each device is identified by its logical name, as shown below:



Creation Properties

Property Name	Property Value
logicalName	
serviceInstanceFacto...	
serviceClass	

JavaPOS Properties

Property Name	Property Value
jposVersion	
deviceCategory	

Vendor Properties

Property Name	Property Value
vendorName	
vendorURL	

Product Properties

Property Name	Property Value
productName	
productURL	
productDescription	

Messages

```
Top of FileLoaded current JposEntryRegistry with 37 entry(ies)
```

Most of the properties of predefined Datalogic devices will never need to be changed. One notable exception is the COM port assignments for RS–232 devices, which have to be modified if they are different from the default value. To change the COM port for an RS–232 device, perform the following steps:

1. Select the device's logical name (for example, `Datalogic-6000-RS-232-Scanner` for a scanner device).
2. Select the **Bus Properties** tab. Click **Edit**.
3. In the **portName** field, type the number of the correct COM port. Click **OK**.
4. Pull down the **File** menu and select **Save JposEntryRegistry**. The new value is saved.
5. Close the JER Editor.

Technical Details

Support Jars

The installation procedure for these drivers provides everything that is required to completely run the drivers with a standalone test application that uses JavaPOS. This may mean that on an actual POS system, some of these files may already exist in different folders. There is always a possibility that conflicts could result because of this, so it is important to remember that the `CLASSPATH` and `PATH` required by the Datalogic drivers is appended to the end of the current `CLASSPATH` and `PATH`. If there are older versions of some of these jar files in the `CLASSPATH` ahead of the ones required by the Datalogic drivers, it may be necessary to rearrange the `CLASSPATH` accordingly. Every situation will have to be looked at uniquely.

The Datalogic drivers were written to the JavaPOS 1.13 specification, so some of the support jar files are a result of the runtimes required to make this function under JavaPOS 1.14. Other support jar files are used for RS-232 and USB communications. If conflicts result it may be important to know what these files are. This is a list of files that get copied, and to which “package” they belong:

```
“.” = Install PATH
```

“Base Datalogic Drivers”

```
   ./JavaPOS
   ./dls.properties
   ./brand.properties
   ./visable.panels.properties
   ./SupportJars/cmdfw.jar
   ./SupportJars/jargs.jar
   ./SupportJars/commons-lang-2.6.jar
   ./LabelIdentifiers.csv
   ./SupportJars/log4j2.xml
   ./SupportJars/log4j-api-2.3.jar
   ./SupportJars/log4j-core-2.3.jar
   ./jpos.xml
```

“javax.comm. Libraries”

```
   ./SupportJars/nrjavaserial-3.9.3.jar
   ./SupportJars/javax.comm.properties
   ./SupportJars/comm.jar
   ./SupportJars/win32com.dll (Windows ONLY)
   ./SupportJars/libLinuxSerialParallel.so (Linux ONLY)
   ./SupportJars/libLinuxSerialParallel_g.so (Linux ONLY)
```

“javax.usb Libraries”

```
   ./SupportJars/javax.usb.properties
   ./SupportJars/jsr80_linux.jar
   ./SupportJars/jsr80_ri.jar
   ./SupportJars/jsr80.jar
    ./SupportJars/libJavaxUsb32.so (32-bit Linux ONLY)
   ./SupportJars/libJavaxUsb64.so (64-bit Linux ONLY)
   ./SupportJars/Win32Hid.dll (32-bit Windows ONLY)
   ./SupportJars/Win32Hid.pdb (32-bit Windows ONLY)
   ./SupportJars/Win64Hid.dll (64-bit Windows ONLY)
   ./SupportJars/Win64Hid.pdb (64-bit Windows ONLY)
```

"POSTest"

```
/SupportJars/POStest.jar
./SupportJars/appframework-1.0.3.jar
./SupportJars/beansbinding-1.2.1.jar
./SupportJars/swing-layout-1.0.3.jar
./SupportJars/swing-layout-1.0.4.jar
./SupportJars/swing-worker-1.1.jar
./POSTest.bat (Windows ONLY)
./POSTest.sh (Linux ONLY)
./POSTest_with_JMX_support.sh (Linux ONLY)
./POSTest.csh (Linux ONLY)
./JEREditor.bat (Windows ONLY)
./JEREditor.sh (Linux ONLY)
./JEREditor.csh (Windows ONLY)
./install.bat (Windows ONLY)
./install.sh (Linux ONLY)
./install.csh (Linux ONLY)
./cmdfwc.bat (Windows ONLY)
./cmdfwc.sh (Linux ONLY)
./cmdfwc.csh (Linux ONLY)
./cmdfw.sh (Linux ONLY)
./cmdfwconf (Linux ONLY)
./cmdfwi.bat (Windows ONLY)
./cmdfwi.sh (Linux ONLY)
./cmdfwi.csh (Linux ONLY)
./cmdfwl.bat (Windows ONLY)
./cmdfwl.sh (Linux ONLY)
./cmdfwl.csh (Linux ONLY)
./cmdfwu.bat (Windows ONLY)
./cmdfwu.sh (Linux ONLY)
./cmdfwu.csh (Linux ONLY)
./test/readme.txt
./test/VideoControls.java
./test/EnterKeyTest.java
```

"javapos.com"

```
./SupportJars/jcl.jar
./SupportJars/jcl_editor.jar
./SupportJars/jpos.properties
./SupportJars/jpos114-controls.jar
./SupportJars/jpos114.jar
./SupportJars/xercesImpl.jar
./SupportJars/xml-apis.jar
./SupportJars/jna-4.0.0.jar
./SupportJars/jna-min-4.0.0.jar
./SupportJars/jna-platform-4.0.0.jar
```

“Update CLASSPATH”

Windows

A new system environment variable DLS_CLASSPATH is created that contains the following string:

```
C:\Program Files\Datalogic\JavaPOS;C:\Program
Files\Datalogic\JavaPOS\SupportJars;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jargs.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\commons-lang-
2.6.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\cmdfw.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\log4j-api-
2.3.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\log4j-core-
2.3.jar;C:\Program
Files\Datalogic\JavaPOS\JavaPOS.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\comm.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\nrjavaserial-
3.9.3.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jsr80_ri.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jsr80.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\appframework-
1.0.3.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\swing-layout-
1.0.4.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\swing-worker-
1.1.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\swing-layout-
1.0.3.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\POSTest.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\beansbinding-
1.2.1.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\xml-apis.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jna-platform-
4.0.0.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\xercesImpl.jar;C:\Progra
m Files\Datalogic\JavaPOS\SupportJars\jpos114-
controls.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jcl.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jna-min-
4.0.0.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jpos114.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jna-4.0.0.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jcl_editor.jar;
```

This environment variable is then appended to the end of the system environment variable “CLASSPATH”.

“Update COM port Access”- LINUX ONLY

Under Linux there are a couple of changes that need to be done to the com ports in order for the Datalogic serial devices to function properly.

All users must be granted access to use the COM ports.

By default the installation program will only change the authorization for the first four COM ports on the system. This is done by adding lines to the /etc/rc.d/rc.local or /etc/rc.d/boot.local file.

```
chmod a+rw /dev/ttyS0
chmod a+rw /dev/ttyS1
chmod a+rw /dev/ttyS2
chmod a+rw /dev/ttyS3
```

If need be, these changes can be manually undone after the installation process has completed. It is also just as easy to add more COM ports to the list, if required.

The COM ports are changed to “raw” mode.

The default behavior for com ports under Linux is for them to act as remote terminals. This means that certain binary data is converted — this needs to be turned off. The first four COM ports are changed to “raw” mode in the /etc/rc.d/rc.local or /etc/rc.d/boot.local file.

```
stty -file=/dev/ttyS0 raw
stty -file=/dev/ttyS1 raw
stty -file=/dev/ttyS2 raw
stty -file=/dev/ttyS3 raw
```

If additional COM ports need to be turned into raw mode, or some need to be changed back to non raw mode, the `/etc/rc.d/rc.local` (fedora) or `/etc/rc.d/boot.local` (SuSE) file can be edited as required.

”Update USB Access”- LINUX ONLY

Under Fedora and SuSE the default behavior is to grant limited access to the USB ports. The access to these ports is changed by modifying the `/etc/fstab` file under SuSE, and the `/etc/rc.d/rc.sysinit` file under Fedora.

SuSE

The following line in the /etc/fstab file:

```
usbfs     /proc/bus/usb     usbfs        noauto 0 0
```

is changed to:

```
usbfs    /proc/bus/usb    usbfs       noauto,devmode=0666
0 0
```

Or for SuSE 11 SP2 and other Linux distributions that use Kernel versions greater than 2.6.31-18

```
/sys/bus/usb/drivers   /proc/bus/usb  usbfs
auto.devmode=0666 0 0
```

Fedora

The following line in the `/etc/rc.d/rc.sysinit` file:

```
[ -d /proc/bus/usb ] && mount -n -t usbfs /proc/bus/usb /
proc/bus/usb
```

is changed to:

```
[ -d /proc/bus/usb ] && mount -n -t usbfs -o devmode=0666 /
proc/bus/usb /proc/bus/usb
```

”Update LD_LIBRARY_PATH”- LINUX ONLY

The `LD_LIBRARY_PATH` variable needs to be defined because some of the low level Java IO uses JNI to talk to the devices. The `LD_LIBRARY_PATH` helps Java find any shared objects that need to be loaded at the appropriate times. A new variable `DLS_LD_LIBRARY_PATH` is created in the `/etc/profile` file, then appended to the `LD_LIBRARY_PATH` variable, as in the following example lines:

```
DLS_LD_LIBRARY_PATH=/opt/dls/JavaPOS/SupportJars
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DLS_LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

NOTE

Any JavaPOS app must use the java command line option:
`-Djava.library.path=/opt/dls/JavaPOS/SupportJars`
to ensure proper libraries are used

PATH — WINDOWS ONLY

During the installation process a new system environment variable by the name of `DLS_PATH` is created that contains the following string:

```
c:\Program Files\Datalogic\JavaPOS;c:\Program
Files\Datalogic\JavaPOS\SupportJars.
```

This environment variable is then appended to end of the system environment variable `PATH`. This is required so the runtime libraries of `Win32Hid.dll` and `Win64Hid.dll` will be found.

Device Service Control

The Datalogic Device Controls use a properties file to control certain optional operations. This file is named `dls.properties`. Upon initial service startup, if the service does not find the `dls.properties` file, it extracts a default version from the JAR file and places it in the same folder as the Java application started from. The user can now change the default settings, but the file must remain in the application directory.

dls.properties file

The following are the contents of the properties file and their descriptions.

Property: `HDLRecordRetry`
Default Value: `3`
Description: Denotes the number of times the scanner update process will resend a failed record before the process will exit and record a failure.

Property: `HDLResetWaitTime`
Default Value: `30`
Description: Denotes the number of seconds that the scanner update process will wait before attempting to communicate with the device to determine success after the reset has been issued. The value may not be set lower than 25 seconds, JavaPOS will use 25 seconds as the minimum. If the update process is being invoked on a mulit-plane fixed scanner using the USB COM interface on a Linux OS, then this value may need to be increased to a minimum of 45 seconds to accurately denote success/failure when returning the device to its previous service state.

Property: `TempDir`
Default Value: `%Temp%` (Windows), `/tmp` (Linux)
Description: Denotes the target directory for any files generated by the service control such as info files, service logs, etc.

Property: `com.dls.jpos.common.ScannerService.AvalancheEnabled`
Default Value: `true`
Description: If set to true, a .prf file is created for use with Avalanche

Property: `com.dls.jpos.common.ScannerService.ScannerInfoFilename`
Default Value: `info`
Description: Used as the suffix to the information file. The device class (from jpos.xml) is prefixed to this name. Example RS232Scanner_<property_value>.txt

Property: `com.dls.jpos.common.ScannerService.generateStatsOnEveryClaim`
Default Value: `true`
Description: If set to true, i-h-s data will be populated and written to the info file on every claim (internal WMI/MBeans data will also be populated). If set to false, the data will only be populated on first claim.

Property: `com.dls.jpos.interpretation.autoLoadConfig`
Default Value: `true`
Description: If set to true, the configuration items available to the USB OEM interface are read for the JavaPOS registry and sent to the attached scanner/ scale.

*Scanners will also need the `configOnClaim` registry entry set to `true`.

Property: `com.dls.jpos.service.ScaleService.ThrowExceptionOnScaleMotion`
Default Value: `false`
Description: If set to true, the service will throw an exception when a weight request times out due to Scale In Motion (112) or Scale At Zero (114). If set to false, the service reports a timeout.

Property: `com.dls.jpos.service.ScannerService.convertBCDtoASCII`
Default Value: `true`
Description: When true the data from the scanner is converted to ASCII for numbers 0 thru 9

Property: `com.dls.jpos.common.ScannerService.generateStatsOnEveryClaim`
Default Value: `true`
Description: When true every time the Claim method is called the i, h, s com- mands are sent to the attached device and responses are parsed and the internal WMI, MBeans data is populated. If false then the WMI and MBeans data is populated on the first Claim.

Property: `com.dls.jpos.service.ScannerService.sendCookedData`
Default Value: `false`
Description: When true the data that is in the scanDataLabel property is copied into the scanData property

The two remaining entries are described here:

`com.Datalogic.jpos.service.ScaleService.ThrowExceptionOnScaleMotion`

Defaults to `False`. If set `True`, the service will throw an exception when a weight request times out due to `Scale In Motion` (112) or `Scale At Zero` (114). If set `False`, the service reports a timeout.

`com.Datalogic.jpos.interpretation.autoLoadConfig`

Default is `True`. When `True`, the service will attempt to configure the scanner/scale as per settings in the `jpos.xml` file upon startup. If `False`, the service will not send configuration data to the scanner/scale upon startup.

NOTE

This must be set `True` if `jpos.xml` settings are changed by the user.

Device Service Logging and Debug Capabilities

JavaPOS proves the ability to log events and errors that occur on a JavaPOS system. Events are logged to a configurable log file in a configurable format and at a variety of logging levels.

The DataLogic Device Services provide logging through the Log4j logging package. Information on Log4j can be obtained at the following web address: http://logging.apache.org/log4j/2.x/

Device Service Logging

Logging control

Logging is controlled through the use of a specially formatted xml file that is located in the SupportJars folder of JavaPOS, named **log4j2.xml.** The provided file configures logging to use the system temporary folder as the location for log files and configures a readable format for the logged information. Configuration of the logging can be altered to suit the user's needs. Documentation for configuring the logging can be referenced at http://logging.apache.org/log4j/2.x/manual/configuration.html.

Standard JavaPOS Scanner Operations

See the *Unified POS Retail Peripheral Architecture v1.14* for details on Properties and Methods

Power Reporting Property

Scanner `CapPowerReporting` is STANDARD. Power reporting is supported by some USB COM Datalogic scanners.

Check Health Method

The scanner supports health check. Possible replies include:

- `Internal Hcheck: Successful`
- `Internal Hcheck: Not successful, <subsystem> Fail;`

Subsystems may include the following:

NOTE **The Scanner is model dependent, not all scanners have all the following subsystems**

- Motor
- Horizontal Laser
- Vertical Laser
- Internal Scale
- Remote Display
- EAS system

`Interactive Hcheck: Complete`

Bar Code Data

ScanData property

Always has bar code data when `DataEvent` is fired.

ScanDataLabel property

- Contains data if `DecodeData = True`
- Computes check digit for UPC labels if not present from scanner.

ScanDataType property

- Contains label type if `DecodeData = True`
- Label type as reported on scanner interface – the scanner assigns a label type identifier and sends this with the label data across the OEM USB and RS-232 Single Cable interfaces. The Device Service translates this into one of the UPOS defined label types.

Statistics

CapStatisticsReporting Property = True

Scanner supports `retrieveStatistics` method. This method returns a pointer to an XML data buffer.

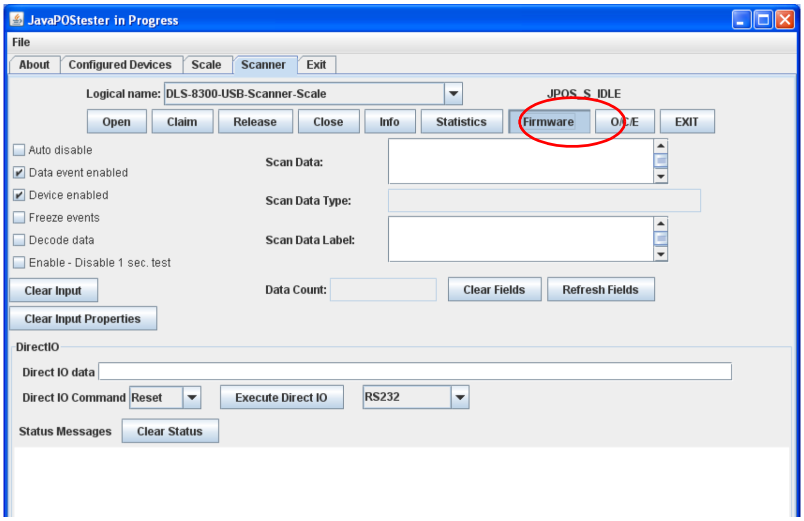
Supported statistics are:

- Device category
- Manufacturer
- Model number
- Serial Number
- Firmware revision
- Interface type
- Power on time
- Number of label scans

CapUpdateStatistics = False

Statistics cannot be updated or reset. Datalogic scanners do not support the `ResetStatistics( )` or `UpdateStatistics( )` methods.

Firmware Update



JavaPOStester in Progress

Logical name: DLS-8300-USB-Scanner-Scale

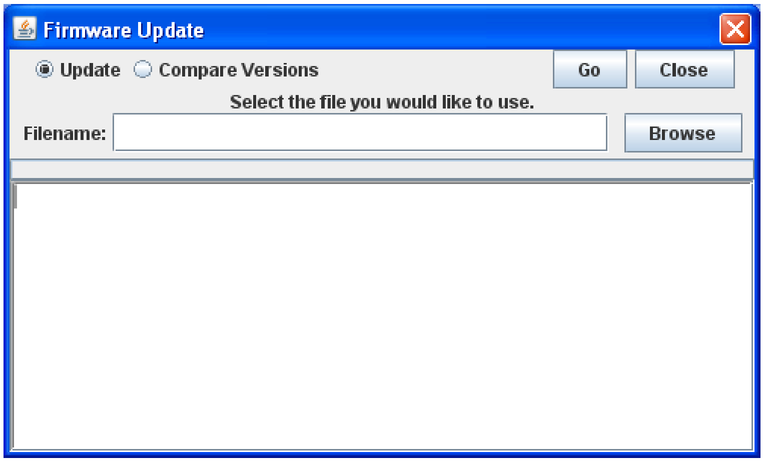
JPOS_S_IDLE

☐ Auto disable
☑ Data event enabled
☑ Device enabled
☐ Freeze events
☐ Decode data
☐ Enable - Disable 1 sec. test

Direct IO Command: Reset

RS232

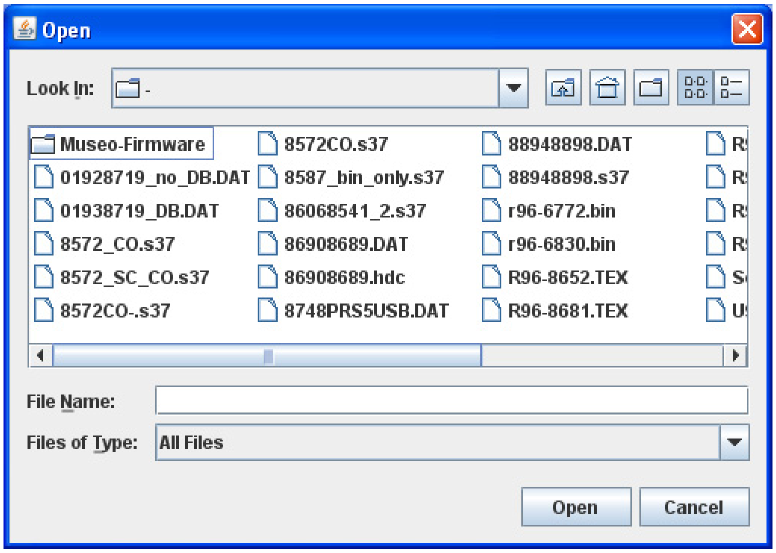
Upon selecting the “Firmware” update button, the following dialog will be presented. The user can then browse to the required update firmware. There are two separate firmware file types: USB or RS-232. Each has the .DAT or .S37 extension, respectively.



Firmware Update

● Update ○ Compare Versions

Select the file you would like to use.



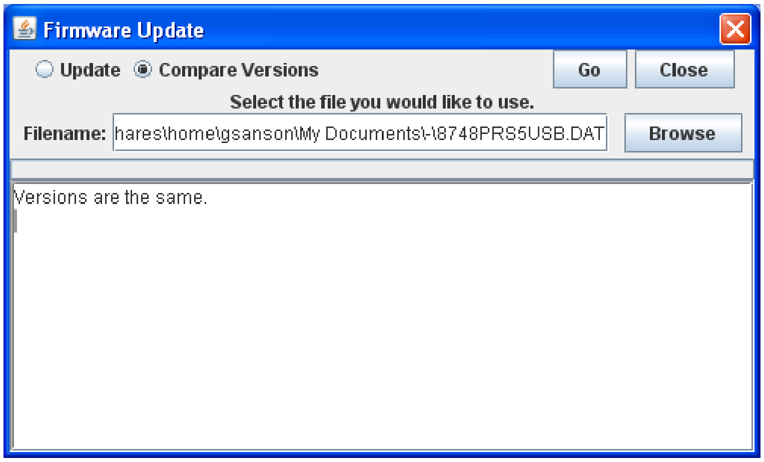
Open

Look In: -

Museo-Firmware
01928719_no_DB.DAT
01938719_DB.DAT
8572_CO.s37
8572_SC_CO.s37
8572CO-.s37
8572CO.s37
8587_bin_only.s37
86068541_2.s37
86908689.DAT
86908689.hdc
8748PRS5USB.DAT
88948898.DAT
88948898.s37
r96-6772.bin
r96-6830.bin
R96-8652.TEX
R96-8681.TEX

Files of Type: All Files

The dialog box will let the user compare or update the firmware. The following is a screen shot of a successful compare when the versions are equal.



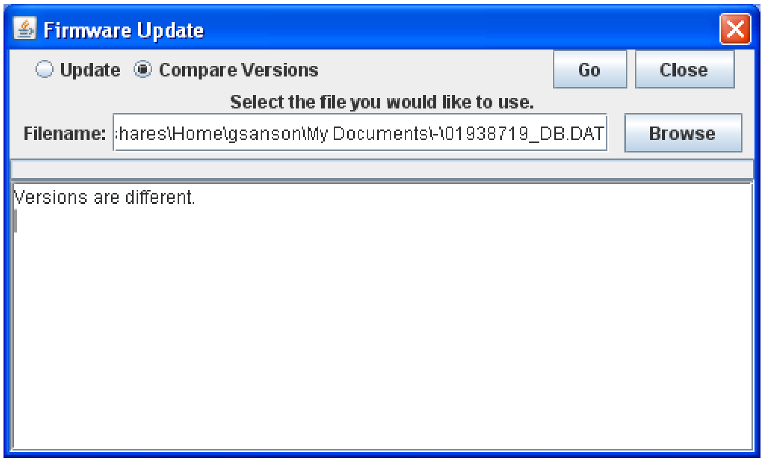
○ Update ● Compare Versions

Select the file you would like to use.

Filename: hares\home\gsanson\My Documents\-\8748PRS5USB.DAT

Versions are the same.

The following is a screen shot of a successful compare when the versions are different.



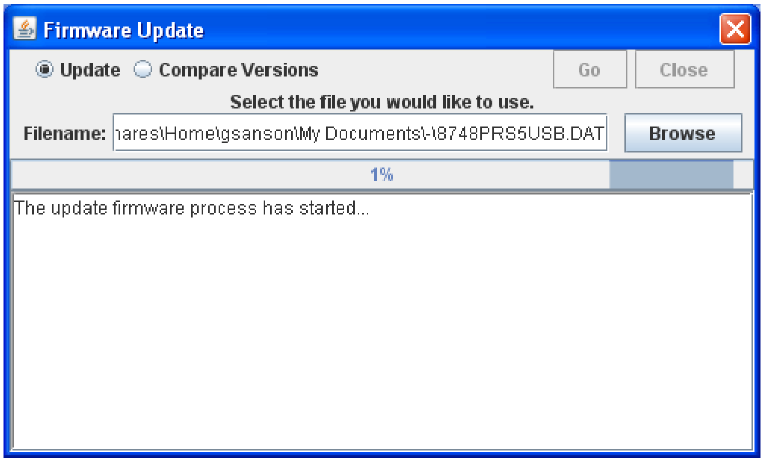
○ Update ● Compare Versions

Select the file you would like to use.

Filename: :hares\Home\gsanson\My Documents\-\01938719_DB.DAT

Versions are different.

This is a screen shot of the firmware update in progress. The status indicator is updated periodically to inform the user of the percentage of update completed.



● Update ○ Compare Versions

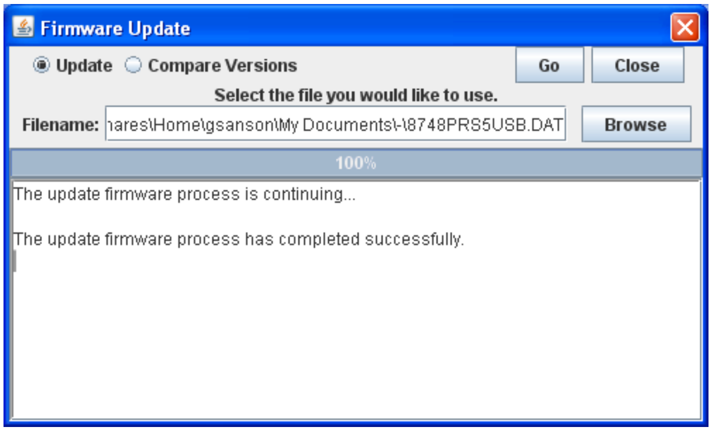
Select the file you would like to use.

Filename: ıares\Home\gsanson\My Documents\-\8748PRS5USB.DAT

1%

The update firmware process has started...

Below is an example of a successfully completed firmware update.



● Update ○ Compare Versions

Select the file you would like to use.

Filename: ıares\Home\gsanson\My Documents\-\8748PRS5USB.DAT

The update firmware process is continuing...

The update firmware process has completed successfully.

Standard JavaPOS Scale operations

See the *Unified POS Retail Peripheral Architecture v1.14* for Properties and Methods.

Power Reporting Property

Scale `CapPowerReporting` is `JavaPOS_PR_NONE:0`. Power reporting is not supported by Datalogic scanner/scales.

The `PowerNotify` property is not used, and changes to it are ignored by Datalogic JavaPOS Device Services. `StatusUpdate` events are not fired by the Datalogic JavaPOS Device Services.

Check Health Method

The scale supports health check, included as part of scanner response. There is no separate health check for scanner or scale. The command is supported on a scale as long as it is claimed, but always reports back a `ResultCode` of “`JavaPOS_SUCCESS`”.

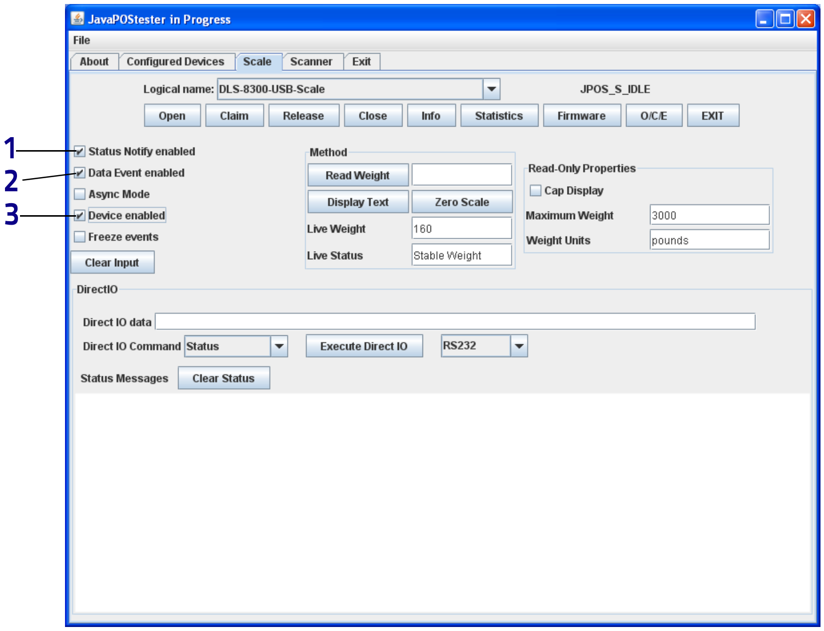
Display

Scale supports a remote display, but does not support text display. The POS application cannot send data to the scale’s remote display (`CapDisplayText = FALSE`).

Live Weight Display

The live weight display can be enabled by selecting the following check boxes in this specific order:

1. Check “Status Notify enabled.”
2. Check “Data Event enabled.”
3. Check “Device enabled.”



Logical name: DLS-8300-USB-Scale

JPOS_S_IDLE

☑ Status Notify enabled
☑ Data Event enabled
□ Async Mode
☑ Device enabled
□ Freeze events

Live Weight: 160
Live Status: Stable Weight

□ Cap Display
Maximum Weight: 3000
Weight Units: pounds

Direct IO data:
Direct IO Command: Status
RS232

After this sequence has been performed, the Live Weight and Live Status fields in the Scale tab will be automatically updated as items are added or removed from the attached scale.

Scanner Direct IO Commands

The scanner supports a number of Direct IO commands that allow access to the commands available on the interface which are not directly supported by UPOS. Applications may utilize the Direct IO commands by accessing the Direct IO Method (see the Unified POS Retail Peripheral Architecture, version 1.14 for details) with the appropriate arguments, as detailed below. For a complete list of commands and implementation details, see DR90000351, Datalogic UPOS DirectIO Commands (available at www.datalogic.com).

Table 1. OEM USB Direct IO Scanner Commands

Scanner Command	Cmd	int[] data	Object object
Reset	1	None	None
Beeper enable	2	None or Input: Config String	Output: Status string
Beeper disable	3	None or Input: Config String	Output: Status string
Configure scanner	4	None or Input: Config String	Input: 9 byte config string Output: Status string
Report configuration	5	None or Input: Config String	Output: 9 byte config string
Configure 2-label flags	6	None or Input: Config String	Input: 8 byte 2-label config string Output: Status string
Report 2-label config	7	None or Input: Config String	Output: 8 byte 2-label config string
Report Information	8	None or Input: Config String	Output: Variable length string
Report Health	9	None or Input: Config String	Output: Variable length string
Report Statistics	10	None or Input: Config String	Output: Variable length string

Table 2. RS-232 Standard Scanner Direct IO Commands

Scanner Command	Cmd	int[] data	Object object
Reset	1	None	None
Beep	11	None	None

Table 3. RS-232 Single Cable Direct IO Scanner/Scale Commands

Scanner Command	Cmd	int[] data	Object object
Hard Reset	21	None	None
Scale Monitor	26	Output: Status	Output: None
Scale Status	27	None	Output: Scale status
Display Status	28	None	Output: Display status
Soft Power Down	29	None	None

Scale Direct IO Commands

The scale supports a number of Direct IO commands that allow access to the commands available on the OEM/IBM USB interface which are not directly supported by UPOS. Applications may utilize the Direct IO commands by accessing the Direct IO Method with the appropriate arguments, as detailed below. For a complete list of commands and implementation details, see DR90000351, Datalogic UPOS DirectIO Commands (available at www.datalogic.com).

Table 4. OEM/IBM USB Direct IO Scale Commands

Scale Command	Cmd	int[] data	Object object
Reset	21	None	None
Enable extended status	22	Output: 3 byte status	Output: Status string
Disable extended status	23	Output: 2 byte status	Output: Status string
Configure scale	24	Output: 2 or 3 byte status	Input: Configuration string Output: Status string
Report configuration	25	Output: 2 or 3 byte status	Output: Configuration string

Scanner and Device Service Configuration

For the JavaPOS device services to work properly with a Datalogic scanner, it is crucial that interface-specific configuration items in the device service and scanner match. This restriction does not apply for the OEM USB interface, as there are no user-configurable items specific to this interface. However, the RS-232 Standard and RS-232 Single Cable have many configuration items that must be coordinated. The listing below for each of these interfaces describes the specific items and their relationships.

Scanner configuration is controlled through the scanner's configuration file, which is loaded into the scanner at the factory. Specific configuration items can be changed by using special programming label barcodes available in the product manual.

Device Service configuration is controlled through the jpos.xml file, accessible in the `Program files\Datalogic\JavaPOS` directory under Windows. Default values are shown in the table below. These values can be edited using the JER Editor to conform to the scanner values.

Table 5. RS-232 Standard Configuration

Scanner configuration item	JavaPOS XML configuration item
Baud rate	Baudrate
Parity	Parity
Data bits	Databits
Stop bits	Stopbits

Table 6. RS-232 Single Cable Configuration

Scanner configuration item	JavaPOS XML configuration item
Baud rate	Baudrate
Parity	Parity
Data bits	Databits
Stop bits	Stopbits
Prefix value, prefix enable/disable	rxPrefix, txPrefix (note: 0 = disable)
Suffix value, suffix enable/disable	rxTrailer, txTrailer (note: 0 = disable)
BCC	UseBCC

Table 7. Scale Configuration

Scale configuration item	JavaPOS XML configuration item
Baud rate	Baudrate
Parity	Parity
Data bits	Databits
Stop bits	Stopbits

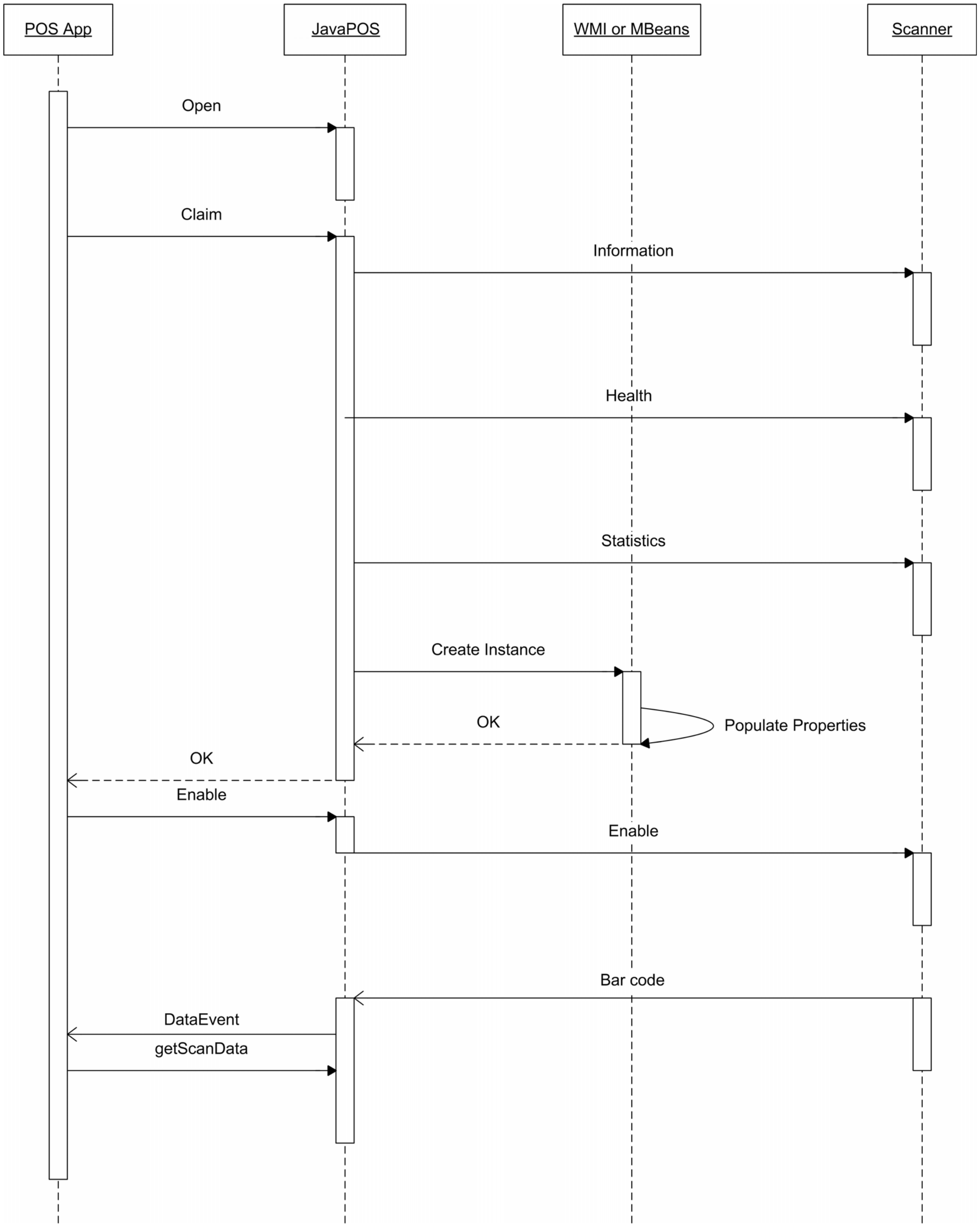
Appendix A. Remote Management Support

This section describes Remote Management support features included in the JavaPOS version 1.14 and above. Remote Management consists of methods to retrieve statistical data about connected devices and the ability to update firmware through the JavaPOS interface. This update feature is also known as Host Download.

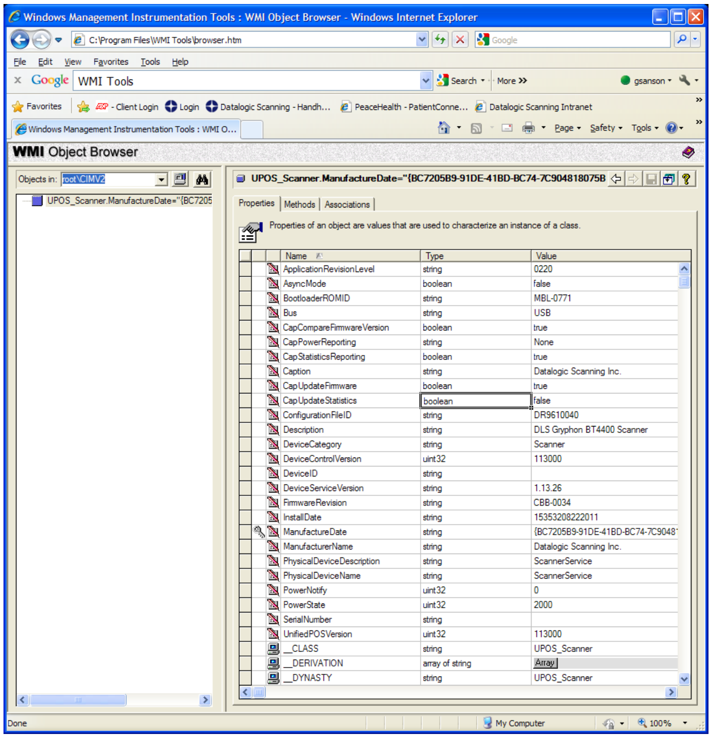
The current methods for statistical data retrieval use either the WMI (Windows Management Interface) or the Java JMX (Java Management Extension) using Managed Beans or MBeans. Each interface uses the `root/CIMV2/UPOS_Scanner` or `root/CIMV2/UPOS_Scale` structure. The following screen shots show the WMI & MBeans data using the Windows Management Information Tools and jconsole, respectively.

Sequence Diagram

The following diagram portrays the normal process with which the WMI or MBeans properties are populated.



WMI Browser


Windows Management Instrumentation Tools : WMI Object Browser - Windows Internet Explorer
C:\Program Files\WMI Tools\browser.htm
Google
File Edit View Favorites Tools Help
Google WMI Tools
Search More »
gsanson
Favorites
- Client Login
Login
Datalogic Scanning - Handh...
PeaceHealth - PatientConne...
Datalogic Scanning Intranet
Windows Management Instrumentation Tools : WMI O...
Page Safety Tools
WMI Object Browser
Properties
Methods
Associations
Done
My Computer
100%

Objects in: root\CIMV2

UPOS_Scanner.ManufactureDate="{BC7205

UPOS_Scanner.ManufactureDate="{BC7205B9-91DE-41BD-BC74-7C904818075B

Properties of an object are values that are used to characterize an instance of a class.

Name	Type	Value
ApplicationRevisionLevel	string	0220
AsyncMode	boolean	false
BootloaderROMID	string	MBL-0771
Bus	string	USB
CapCompareFirmwareVersion	boolean	true
CapPowerReporting	string	None
CapStatisticsReporting	boolean	true
Caption	string	Datalogic Scanning Inc.
CapUpdateFirmware	boolean	true
CapUpdateStatistics	boolean	false
ConfigurationFileID	string	DR9610040
Description	string	DLS Gryphon BT4400 Scanner
DeviceCategory	string	Scanner
DeviceControlVersion	uint32	113000
DeviceID	string	
DeviceServiceVersion	string	1.13.26
FirmwareRevision	string	CBB-0034
InstallDate	string	15353208222011
ManufactureDate	string	{BC7205B9-91DE-41BD-BC74-7C9048
ManufacturerName	string	Datalogic Scanning Inc.
PhysicalDeviceDescription	string	ScannerService
PhysicalDeviceName	string	ScannerService
PowerNotify	uint32	0
PowerState	uint32	2000
SerialNumber	string	
UnifiedPOSVersion	uint32	113000
__CLASS	string	UPOS_Scanner
__DERIVATION	array of string	Array
__DYNASTY	string	UPOS_Scanner

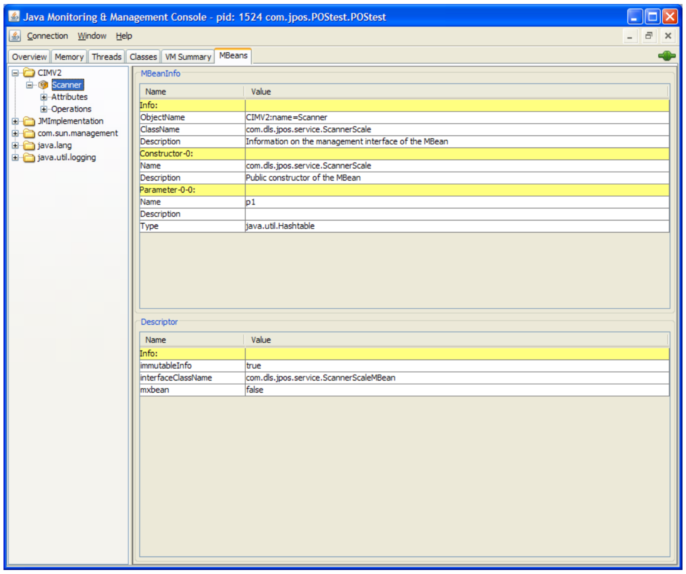
JMX MBeans Browser using jconsole

The following are screen shots of the jconsole application supplied with the JDK 1.5 or above. The JavaPOS application must be started with the option:

```
-Dcom.sun.management.jmxremote
```

so that the JVM will accept and reply to remote management calls.

JMX MBeans Browser using jconsole



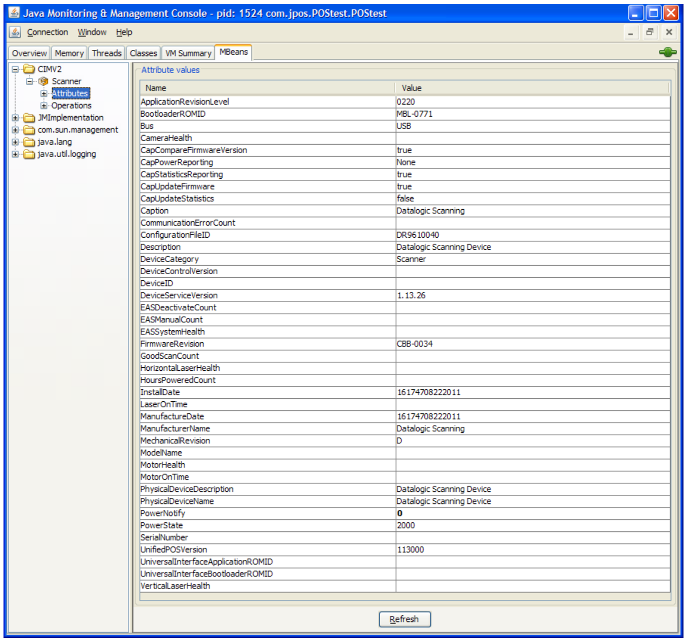
MBeanInfo

Name	Value
Info:	
ObjectName	CIMV2:name=Scanner
ClassName	com.dls.jpos.service.ScannerScale
Description	Information on the management interface of the MBean
Constructor-0:	
Name	com.dls.jpos.service.ScannerScale
Description	Public constructor of the MBean
Parameter-0-0:	
Name	p1
Description	
Type	java.util.Hashtable

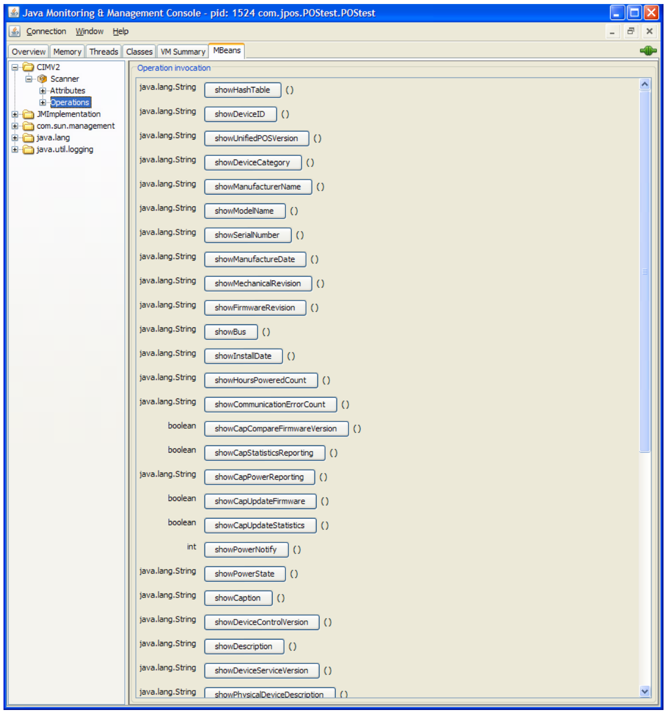
Descriptor

Name	Value
Info:	
immutableInfo	true
interfaceClassName	com.dls.jpos.service.ScannerScaleMBean
mxbean	false

JMX MBeans Browser using jconsole (continued)



Name	Value
ApplicationRevisionLevel	0220
BootloaderROMID	MBL-0771
Bus	USB
CameraHealth	
CapCompareFirmwareVersion	true
CapPowerReporting	None
CapStatisticsReporting	true
CapUpdateFirmware	true
CapUpdateStatistics	false
Caption	Datalogic Scanning
CommunicationErrorCount	
ConfigurationFileID	DR9610040
Description	Datalogic Scanning Device
DeviceCategory	Scanner
DeviceControlVersion	
DeviceID	
DeviceServiceVersion	1.13.26
EASDeactivateCount	
EASManualCount	
EASSystemHealth	
FirmwareRevision	CBB-0034
GoodScanCount	
HorizontalLaserHealth	
HoursPoweredCount	
InstallDate	16174708222011
LaserOnTime	
ManufactureDate	16174708222011
ManufacturerName	Datalogic Scanning
MechanicalRevision	D
ModelName	
MotorHealth	
MotorOnTime	
PhysicalDeviceDescription	Datalogic Scanning Device
PhysicalDeviceName	Datalogic Scanning Device
PowerNotify	**0**
PowerState	2000
SerialNumber	
UnifiedPOSVersion	113000
UniversalInterfaceApplicationROMID	
UniversalInterfaceBootloaderROMID	
VerticalLaserHealth	



java.lang.String	showHashTable ()
java.lang.String	showDeviceID ()
java.lang.String	showUnifiedPOSVersion ()
java.lang.String	showDeviceCategory ()
java.lang.String	showManufacturerName ()
java.lang.String	showModelName ()
java.lang.String	showSerialNumber ()
java.lang.String	showManufactureDate ()
java.lang.String	showMechanicalRevision ()
java.lang.String	showFirmwareRevision ()
java.lang.String	showBus ()
java.lang.String	showInstallDate ()
java.lang.String	showHoursPoweredCount ()
java.lang.String	showCommunicationErrorCount ()
boolean	showCapCompareFirmwareVersion ()
boolean	showCapStatisticsReporting ()
java.lang.String	showCapPowerReporting ()
boolean	showCapUpdateFirmware ()
boolean	showCapUpdateStatistics ()
int	showPowerNotify ()
java.lang.String	showPowerState ()
java.lang.String	showCaption ()
java.lang.String	showDeviceControlVersion ()
java.lang.String	showDescription ()
java.lang.String	showDeviceServiceVersion ()
java.lang.String	showPhysicalDeviceDescription ()

WMI/JMX MBeans Properties

This data is also returned by the `getStatistics` command. Some items are either null or 0 depending on the connected device. Scale-relevant information will not be valid for a scanner-only device. Items listed below are unique to Datalogic's implementation and differ from the properties listed in the UPOS specification. Some of the following properties may be omitted for configurations where they are not applicable.

Property: `DeviceID`

Description: A unique number used to identify the attached device

Device Category: Scanner, Scale, Portal Scanner

Property: `ModelName`

Description: Model number of attached device

Device Category: Scanner, Scale, Portal Scanner

Property: `SerialNumber`

Description: Serial number of attached device

Device Category: Scanner, Scale, Portal Scanner

Property: `ScaleHealth`

Device Category: Scale

Property: `RemoteDisplayHealth`

Device Category: Scale

Property: `ConfigurationFileID`

Device Category: Scanner, Portal Scanner

Property: `ApplicationRevisionLevel`

Device Category: Scanner, Portal Scanner

Property: `BootloaderROMID`

Device Category: Scanner, Portal Scanner

Property: `UniversalInterfaceApplicationROMID`

Device Category: Scanner, Portal Scanner

Property: `UniversalInterfaceBootloaderROMID`

Device Category: Scanner, Portal Scanner

Property: `MotorHealth`

Device Category: Scanner, Portal Scanner

Property: `HorizontalLaserHealth`

Device Category: Scanner, Portal Scanner

Property: `VerticalLaserHealth`

Device Category: Scanner, Portal Scanner

Property: CameraHealth
Device Category: Scanner, Portal Scanner

Property: MotorOnTime
Device Category: Scanner, Portal Scanner

Property: LaserTime
Device Category: Scanner, Portal Scanner

Property: EASSystemHealth
Device Category: Scanner, Portal Scanner

Property: EASDeactivateCount
Device Category: Scanner, Portal Scanner

Property: EASManualCount
Device Category: Scanner, Portal Scanner

Property: EASVersion
Device Category: Scanner, Portal Scanner

In addition to the items above there can also be Hand Held specific information. The following are only available for Hand Held scanners and includes wireless units.

Property: RadiolHealth
Device Category: Scanner

Property: RadiolHardwareVersion
Device Category: Scanner

Property: RadioSoftwareVersion
Device Category: Scanner

Property: BaseSoftwareVersion
Device Category: Scanner

Property: BaseModelNumber
Device Category: Scanner

Property: BaseSerialNumber
Device Category: Scanner

Property: BaseHoursPoweredCount
Device Category: Scanner

Property: BatteryChargeCycles
Device Category: Scanner

Property: `TriggerPulls`
Device Category: Scanner

Property: `KeyPresses`
Device Category: Scanner

Property: `LaserTime`
Device Category: Scanner

Property: `LabelsScanned`
Device Category: Scanner

Avalanche Support

The following are the contents of the scanner_info.txt file as well as the scanner_info.prf file. The scanner_info.prf file is used by Avalanche to populate its database for remote management.

```
# Raw data from scanner
# Info command
# bytes received[87] 0x1 0x2 0x41 0x36 0x31 0x30 0x30
0x30 0x32 0x36 0x30 0x37 0x3 0x2 0x52 0x30 0x32 0x33
0x34 0x3 0x2 0x43 0x44 0x52 0x39 0x36 0x31 0x30 0x58
0x58 0x58 0x20 0x3 0x2 0x42 0x36 0x31 0x30 0x30 0x30
0x32 0x37 0x30 0x35 0x3 0x2 0x53 0x47 0x42 0x30 0x30
0x31 0x34 0x30 0x30 0x36 0x3 0x2 0x4D 0x47 0x42 0x54
0x34 0x34 0x30 0x30 0x2D 0x42 0x4B 0x4B 0x31 0x30 0x2D
0x43 0x31 0x39 0x38 0x3 0x2 0x6D 0x3 0x2 0x49 0x34 0x35
0x3 0x4
# Stats command
# bytes received[65] 0x15 0x50 0x1 0x8 0x0 0x1 0x2 0x4C
0x34 0x32 0x3 0x2 0x43 0x3 0x2 0x50 0x35 0x37 0x35 0x3
0x4 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0
# Health command
# bytes received[65] 0x7 0x50 0x1 0x8 0x0 0x1 0x4 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
0x0

#
```

Below is the same data, displayed differently.

```
# Parsed data
#
Vendor = GBT4400
# UPOS
ManufacturerName = Datalogic ADC INC
DeviceCategory = Scanner
ScaleMode = English
# info
ApplicationROM = 610002607
ApplicationRevisionLevel = 0234
ConfigurationFileID = DR9610XXX
BootloaderROMID = 610002705
SerialNumber = GB0014006
TopModelNumber = GBT4400-BKK10-C198
MainBoardNumber =
Interface = USB-OEM
# health
# statistics
ScanCount = 42
LastScaleCalibrationTime =
PowerOnHours = 575
# Creation date: Wed Sep 05 16:41:45 2012
# Copyright 2012, Datalogic ADC, Inc.
# EOF
```

DLSJavaPOS Properties

The following are the contents of the properties file, dls.properties, and property descriptions.

Property: `com.dls.jpos.service.ScannerService.convertBCDtoASCII`
Default Value: true
Description: When true the data from the scanner is converted to ASCII for numbers 0 thru 9

Property: `com.dls.jpos.interpretation.autoLoadConfig`

Default Value: true

Description: When true the configuration items available to the OEM USB interface are read from the JavaPOS registry and sent to the attached scanner.

Property: `com.dls.jpos.service.ScaleService.ThrowExceptionOnScaleMotion`

Default Value: false

Description: Exception thrown when scale is in motion

Property: `com.dls.jpos.common.ScannerService.AvalancheEnabled`

Default Value: true

Description: When true the .prf file is created for use with Avalanche

Property: `com.dls.jpos.common.ScannerService.generateStatsOnEveryClaim`

Default Value: true

Description: When true every time the Claim method is called the i, h, s commands are sent to the attached device and responses are parsed and the internal WMI, MBeans data is populated. If false then the WMI and MBeans data is populated on the first Claim.

Property: `com.dls.jpos.common.ScannerService.ScannerInfoFilename`

Default Value: scanner_info.txt

Description: used as the suffix to the information file. The device class is prefixed to this name. Example RS232Scanner_scanner_info.txt

Property: `com.dls.jpos.service.ScannerService.sendCookedData`

Default Value: false

Description: When true the data that is in the scanDataLabel property is copied into the scanData property

The following are additional properties:

Property: `com.dls.jpos.service.ScannerService.poleRateForPortChange`

Default Value: 1000

Property: `com.dls.jpos.service.ScannerService.enablePole`

Default Value: false

Property: `com.dls.jpos.service.ScannerService.PostRemovalErrorEvents`

Default Value: false

Property: `com.dls.jpos.service.ScannerService.enablePoleRate`

Default Value: 1

Property: `com.dls.jpos.service.ScannerService.useClaimLockFile`

Default Value: false

Property: `com.dls.jpos.service.ScannerService.suppressErrors`

Default Value: false

Property: `com.dls.jpos.service.ScannerService.FirmwareSendNulls`

Default Value: false

Property: `com.dls.jpos.service.ScannerService.FirmwareSendReset`

Default Value: true

JavaPOS Registry

The contents of the jpos.xml file are used to make decisions on how JavaPOS behaves when the Datalogic devices are attached. The following is a listing of the Datalogic-specific custom properties along with a brief description.

Property: `useVirtualPort`

Default Value: False

Description: When true under Windows causes the USB COM ports to be searched first before using the portName in the jpos.xml file

Property: `deviceClass`

Default Value: RS232Scanner

Description: Used to uniquely identify the scanner info file name and Avalanche prf file name.

Property: `useCOMxOnLinux`

Default Value: False

Description: When true JavaPOS will search for ports with the prefix of COM under the Linux operating system. Example:
COM1 under Linux is `/dev/COM1`

Property: `ipPort`

Default Value: 26666

Description: Port number used to communicate with the PortalScanner

Property: `ipAddress`

Default Value: 192.168.0.2

Description: Address number used to communicate with the PortalScanner

Property: `laneNumber`

Default Value: 0

Description: lane number used to identify location of PortalScanner

Property: `imageBuffers`

Default Value: 1

Description: number of buffers the connected application maintains

Property: `serviceVersion`

Default Value: 1.13.43 (current service version number)

Description: used for messages to and from PortalScanner

Property: `MBeansEnabled`

Default Value: False

Description: Enables the MBeans interface. Must be used with the java virtual machine command line option `-Dcom.sun.management.jmxremote`

Property: `WMIEnabled`

Default Value: False

Description: Enables WMI data store population

Property: `decodeType`

Default Value: standard

Description: Used to determine the label ID sent from the attached scanner. Three types may exist, standard, Warhol/European, or AIM.

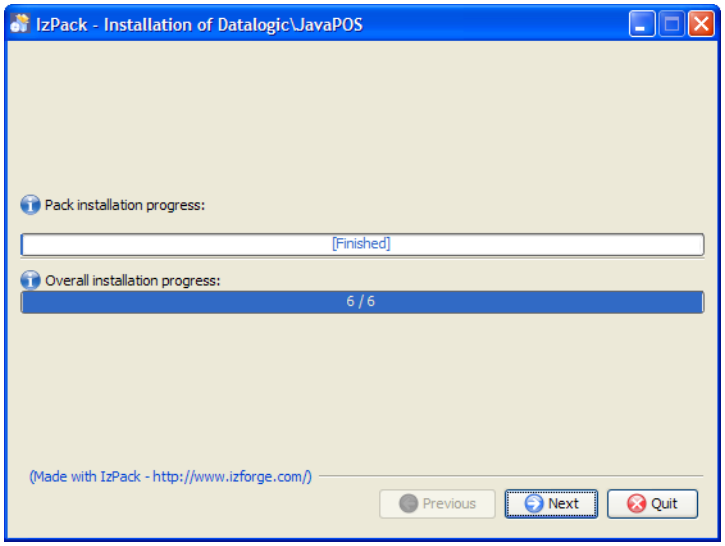
NOTES

Appendix B. Auto Installation Script

Creation and use of auto-installation script

This section describes how to create an auto-installation script for use in unattended installs. The example below uses the standard Windows install directories.

Figure 10. After normal installation is finished



IzPack - Installation of Datalogic\JavaPOS

Pack installation progress:

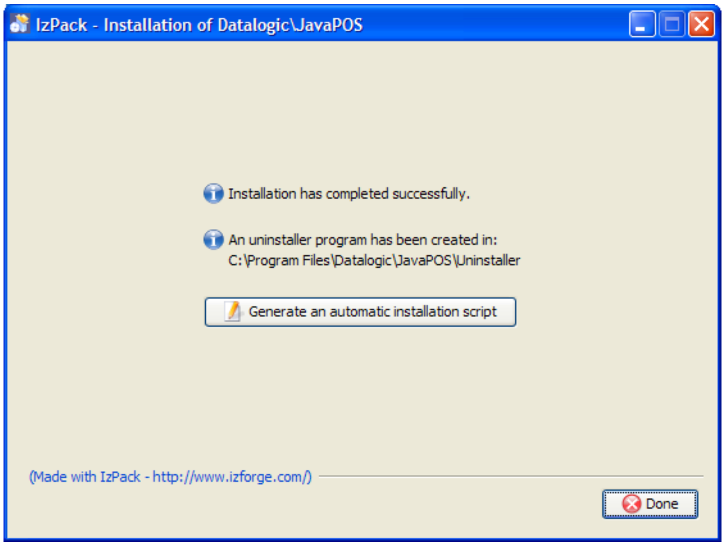
[Finished]

Overall installation progress:

6 / 6

(Made with IzPack - http://www.izforge.com/)

1. Click the "Generate an automatic installation script" button



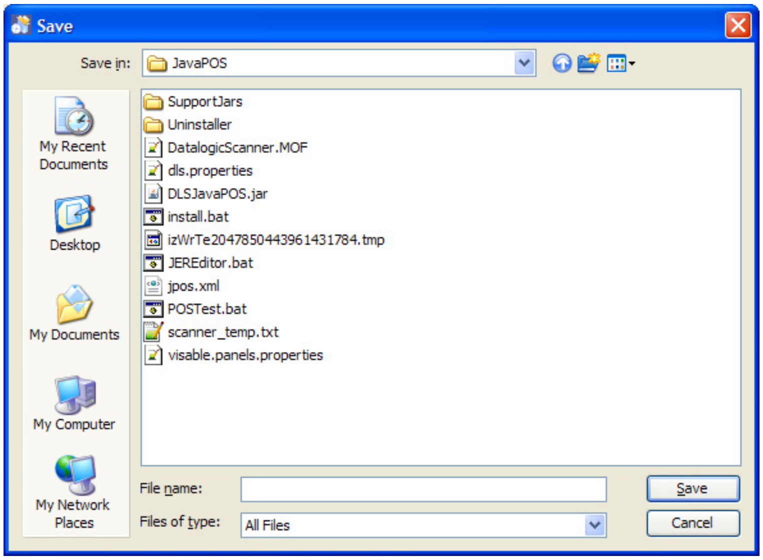
IzPack - Installation of Datalogic\JavaPOS

Installation has completed successfully.

An uninstaller program has been created in:
C:\Program Files\Datalogic\JavaPOS\Uninstaller

Generate an automatic installation script
(Made with IzPack - http://www.izforge.com/)
Done

2. A "Save As" dialog pops up



Save

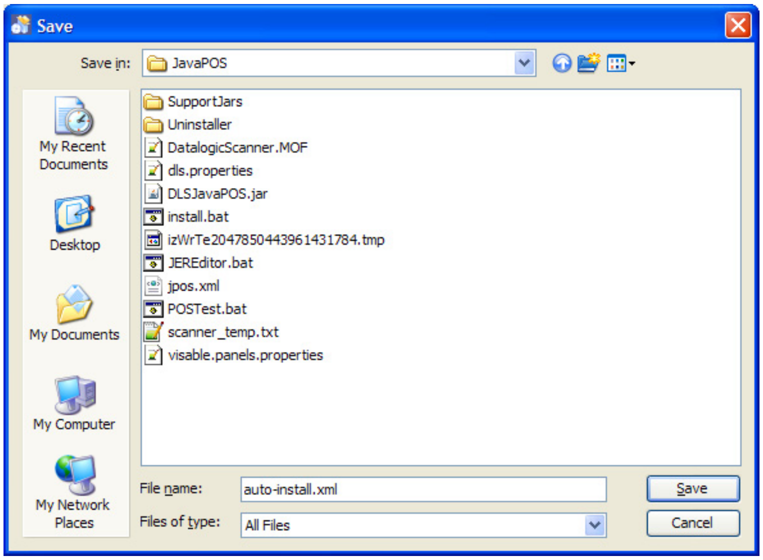
Save in: JavaPOS

```
SupportJars
Uninstaller
DatalogicScanner.MOF
dls.properties
DLSJavaPOS.jar
install.bat
izWrTe2047850443961431784.tmp
JEREditor.bat
jpos.xml
POSTest.bat
scanner_temp.txt
visable.panels.properties
```

File name:

Files of type: All Files

3. Enter "auto-install.xml" as the file name



Save

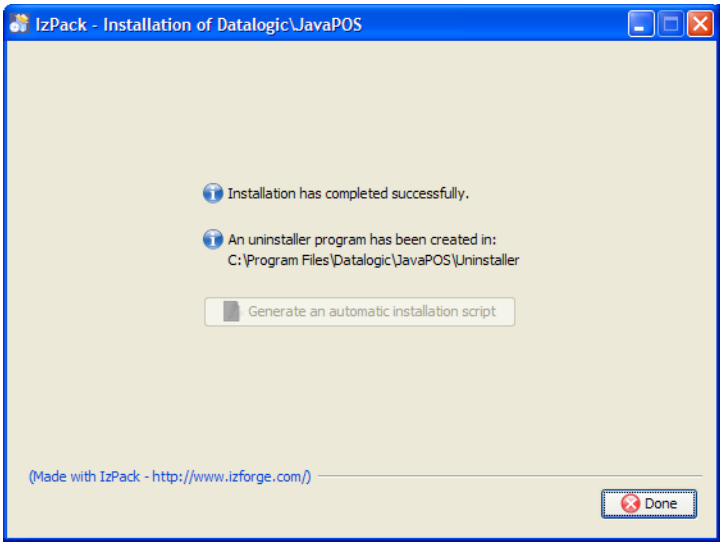
Save in: JavaPOS

SupportJars
Uninstaller
DatalogicScanner.MOF
dls.properties
DLSJavaPOS.jar
install.bat
izWrTe2047850443961431784.tmp
JEREditor.bat
jpos.xml
POSTest.bat
scanner_temp.txt
visable.panels.properties

File name: auto-install.xml

Files of type: All Files

4. Now the file has been created



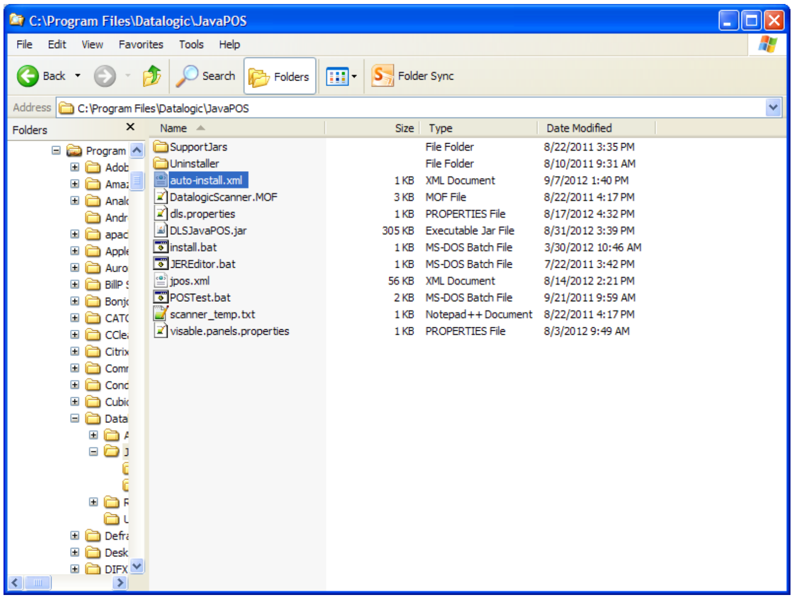
IzPack - Installation of Datalogic\JavaPOS

Installation has completed successfully.

An uninstaller program has been created in:
C:\Program Files\Datalogic\JavaPOS\Uninstaller

(Made with IzPack - http://www.izforge.com/)

5. The "auto-install.xml" file is now located in the installation directory


C:\Program Files\Datalogic\JavaPOS
File Edit View Favorites Tools Help
Back
Search
Folders
Folder Sync
Address C:\Program Files\Datalogic\JavaPOS
Folders
Program

Name	Size	Type	Date Modified
SupportJars		File Folder	8/22/2011 3:35 PM
Uninstaller		File Folder	8/10/2011 9:31 AM
auto-install.xml	1 KB	XML Document	9/7/2012 1:40 PM
DatalogicScanner.MOF	3 KB	MOF File	8/22/2011 4:17 PM
dls.properties	1 KB	PROPERTIES File	8/17/2012 4:32 PM
DLSJavaPOS.jar	305 KB	Executable Jar File	8/31/2012 3:39 PM
install.bat	1 KB	MS-DOS Batch File	3/30/2012 10:46 AM
JEREditor.bat	1 KB	MS-DOS Batch File	7/22/2011 3:42 PM
jpos.xml	56 KB	XML Document	8/14/2012 2:21 PM
POSTest.bat	2 KB	MS-DOS Batch File	9/21/2011 9:59 AM
scanner_temp.txt	1 KB	Notepad++ Document	8/22/2011 4:17 PM
visable.panels.properties	1 KB	PROPERTIES File	8/3/2012 9:49 AM

Use this file along with the **install.bat** and **setup.jar** to perform command line installation, as the contents of the **install.bat** imply.

Contents of install.bat

```
@echo off
java -jar setup.jar auto-install.xml
```

Contents of auto-install.xml file

```
<AutomatedInstallation langpack="eng">
    <com.izforge.izpack.panels.DLSHelloPanel/>
    <com.izforge.izpack.panels.LicencePanel/>
    <com.izforge.izpack.panels.TargetPanel>
        <installpath>C:\Program 
Files\Datalogic\JavaPOS</installpath>
    </com.izforge.izpack.panels.TargetPanel>
    <com.izforge.izpack.panels.PacksPanel>
        <selected>
            <pack index="0"/>
            <pack index="1"/>
            <pack index="2"/>
            <pack index="3"/>
            <pack index="4"/>
            <pack index="5"/>
        </selected>
    </com.izforge.izpack.panels.PacksPanel>
    <com.izforge.izpack.panels.InstallPanel/>
    <com.izforge.izpack.panels.FinishPanel/>
</AutomatedInstallation>
```

Appendix C. JavaPOS Command-Line Host Download Utility

About the Command-Line Host Download Utility

JavaPOS provides the ability to update scanners using a command-line utility that is written in Java and uses the JavaPOS drivers. Shell scripts and batch files are provided below as examples.

Command line invocation:

From a command prompt or shell, the following command may be invoked from the JavaPOS directory after the **CLASSPATH** environment variable has been set to include these jar files:

```
JavaPOS.jar
jargs.jar
jcl.jar
jcl_editor.jar
jna.jar
jpos114.jar
jsr80-1.0.2.5.jar
jsr80_linux-1.0.3.jar
jsr80_ri-1.0.2.5.jar
nrjavaserial-3.9.3.jar
xercesImpl.jar
xml-apis.jar
cmdfw.jar
log4j-api-2.3.jar
log4j-core-2.3.jar
```

Then the Java executable can be invoked with the following command line options:

```
java -Djava.ext.dirs=.:SupportJars -
Djava.library.path=.:SupportJars cmdfw.
Cmdfw -v
-s DLS-Magellan-9800i-USB-Scanner-Scale -c -f
DR9610414_CFG_ONLY.DAT
```

NOTE

Please note the different options for the java executable versus the `cmdfw.CMdfw jar` options!

Help output

cmdfw version: 1.14.026

Usage: **`cmdfw.Cmdfw[-verbose] [-help] [-hilc] -scanner <ScannerName> -filename <fileName>`**

The default behavior is to update the firmware using the –scanner **`<ScannerName> -filename <filename>`** options:

Options:

`-h` –help print this information

`-i` print info to screen and to scannerinfo.txt file, **`-scanner <ScannerName>`** is required

`-c` compare **`<filename>`** with firmware in **`<ScannerName>`**, **`-scanner <ScannerName>`** & **`-filename <fileName>`** are required

`-l` list all available ScannerNames in jpos.xml file

`-v` verbose

Examples

There are 8 shell scripts/batch files that are examples of the info, compare, list, and update options.

Compare:

```
cmdfwc.bat
cmdfwc.sh
```

Info:

```
cmdfwi.bat
cmdfwi.sh
```

List:

```
cmdfwl.bat
cmdfwl.sh
```

Update

```
cmdfwu.bat
cmdfwu.sh
```

©2006 - 2018 Datalogic S.p.A. and/or its affiliates ▪ All rights reserved ▪ Without limiting the rights under copyright, no part of this documentation may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means, or for any purpose, without the express written permission of Datalogic S.p.A. and/or its affiliates ▪ Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S. and the E.U.

DATALOGIC

www.datalogic.com

Datalogic USA, Inc.
959 Terry Street | Eugene | OR 97402 | USA
Telephone: (1) 541-683-5700 | Fax: (1) 541-345-7140

820044114 (Rev. F) March 2018