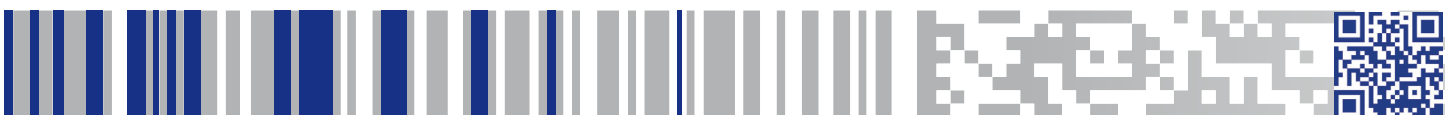




Datalogic JavaPOS Device Services



User Manual

Datalogic USA, Inc.

959 Terry Street
Eugene, OR 97402
USA

Telephone: (541) 683-5700

Fax: (541) 345-7140

©2006 - 2019 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An Unpublished Work - All rights reserved.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

DATALOGIC JAVAPOS DEVICE SERVICES	1
Introduction	1
Document Conventions	1
About Datalogic JavaPOS Device Services	1
Datalogic Products Supported	2
Limitations	2
Installation	2
Windows Installation	3
Linux Installation	7
Verification	8
Example - Configuring a Scanner	9
Technical Details	10
'Update CLASSPATH'	12
SuSE	13
Fedora	13
PATH — WINDOWS ONLY	14
Uninstall	14
Device Service Control	15
dls.properties file	15
Device Service Logging and Debug Capabilities	19
Device Service Logging	19
Logging control	19
Standard JavaPOS Scanner Operations	19
Power Reporting Property	19
Check Health Method	19
Bar Code Data	20
ScanData property	20
ScanDataLabel property	20
ScanDataType property	20
Statistics	20
CapStatisticsReporting Property = True	20
CapUpdateStatistics = False	20
Firmware Update	21
Standard JavaPOS Scale operations	23
Power Reporting Property	23
Check Health Method	23
Display	23
Live Weight Display	24
Standard JavaPOS RFID operations	24
Scanner Direct IO Commands	25
Scale Direct IO Commands	27
Scanner and Device Service Configuration	27
REMOTE MANAGEMENT SUPPORT	29
Sequence Diagram	30
JMX MBeans Browser using jconsole	31
WMI/JMX MBeans Properties	33
Avalanche Support	35
DLSJavaPOS Properties	36

JavaPOS Registry37

AUTO INSTALLATION SCRIPT 39

Creation and use of auto-installation script39

 Command Line Installation41

 Contents of auto-install.xml file (example)42

JAVAPOS COMMAND-LINE

HOST DOWNLOAD UTILITY 43

About the Command-Line Host Download Utility43

 Help output44

 Examples45



Chapter 1

Datalogic JavaPOS Device Services

Introduction

Document Conventions

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



Notes contain additional information of interest to the reader.

Filename, 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 Object Management Group at: <https://www.omg.org/retail/unified-pos>.

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 3.1., 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 7 and later versions.

Installation

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 Vista, Windows 7, Windows 8, Windows 10
- Red Hat Enterprise Linux versions 5+
- SuSE Linux Professional 9.3+, CentOS 6+, OpenSuSE 11+, Ubuntu 16.04+

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 8.
3. ["Technical Details"](#) on page 10 — 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 1.7 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.

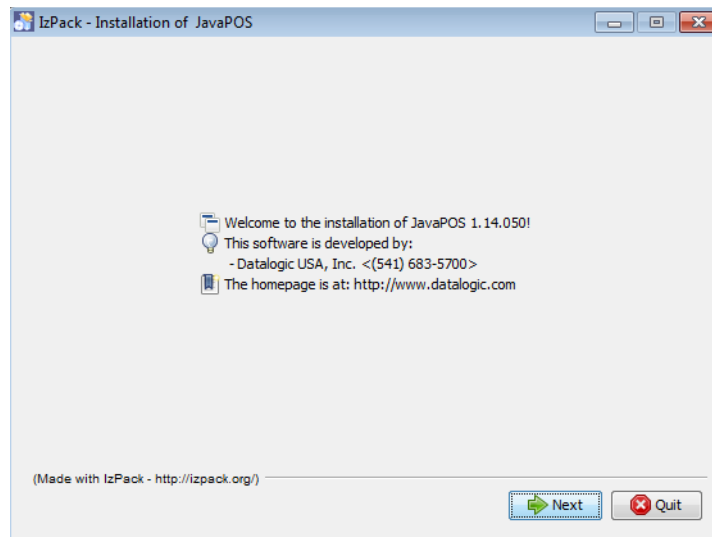
To install Datalogic JavaPOS, perform the following steps:

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

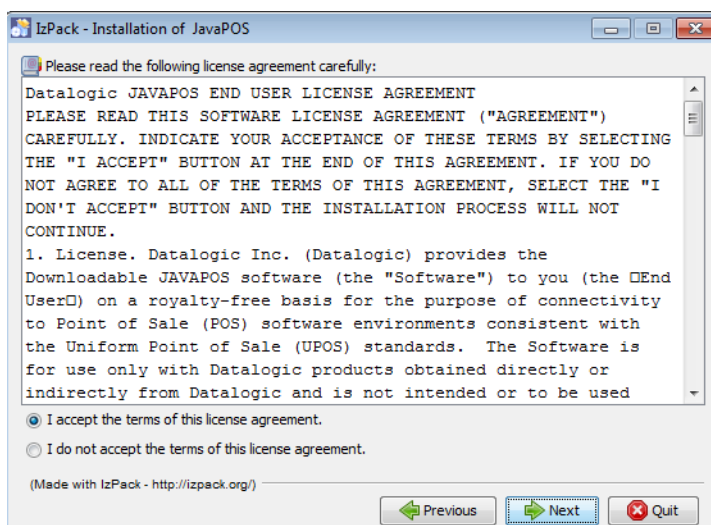


NOTE

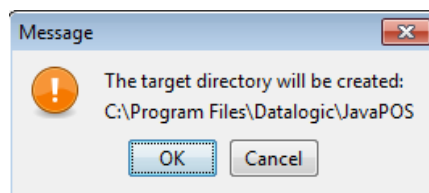
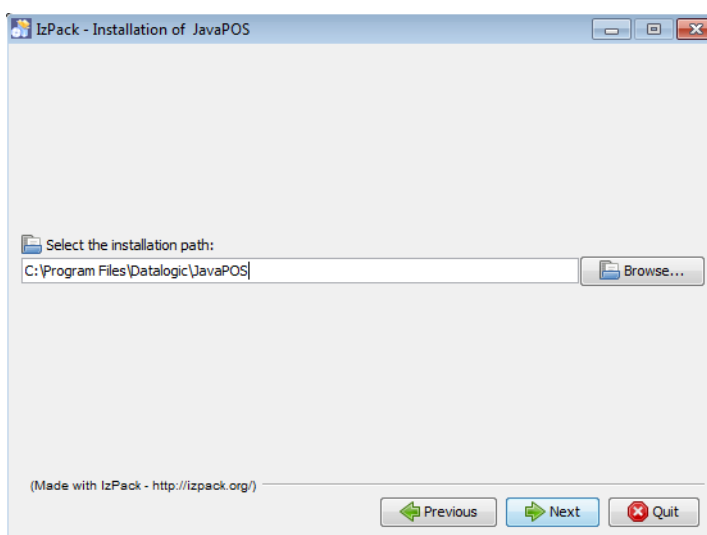
On systems running Windows 7 and later, it may be necessary to install JavaPOS as a user with Administrator privileges.



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.



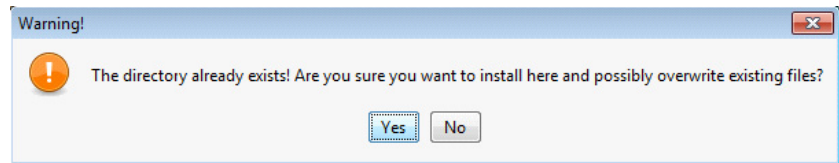
3. Click **Next**. You are prompted to enter an installation path (the default is C:\Program Files\Datalogic\JavaPOS).





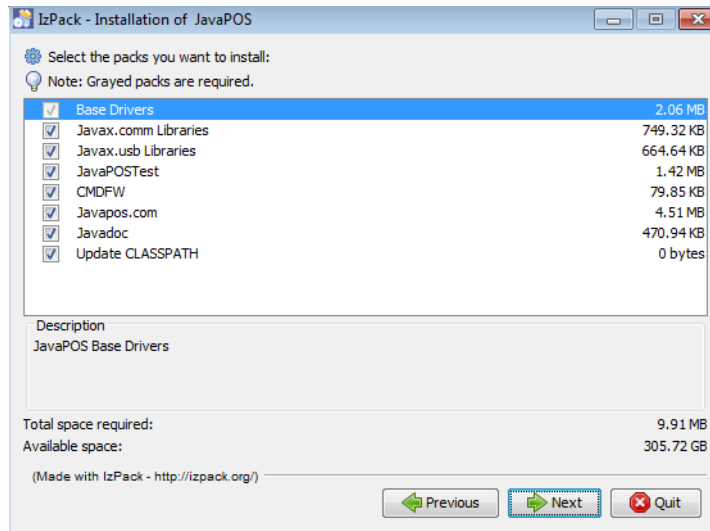
CAUTION

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

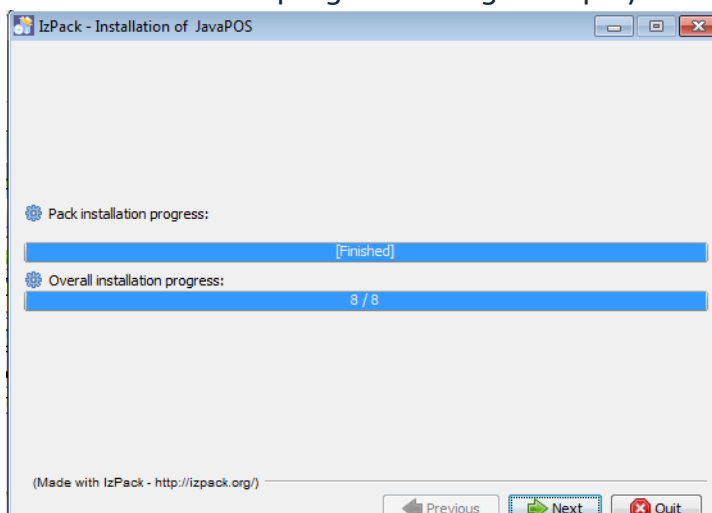


If you receive this message, click **No** to halt the installation. Uninstall any previous version of Datalogic JavaPOS, then return to step 1 of this procedure.

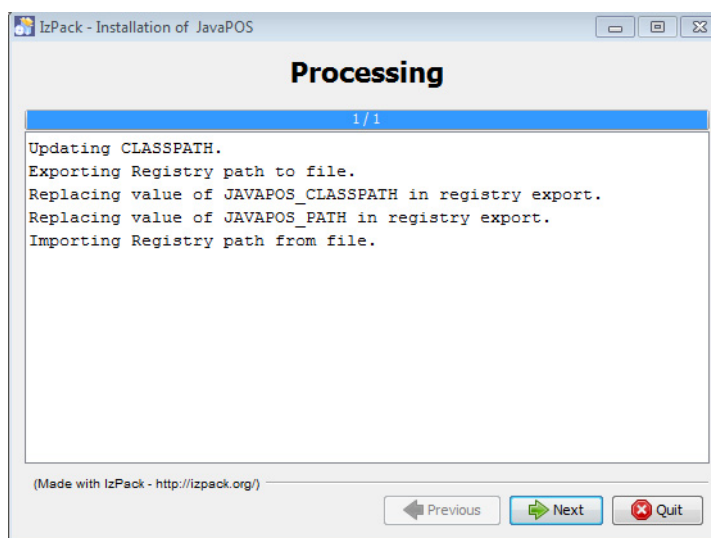
4. Available packages for the current version are displayed. Only the Base Datalogic Drivers are absolutely required. If installing on a system for the first time to test the functionality of the drivers, it is recommended that all packages be installed. It is important to realize that these drivers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



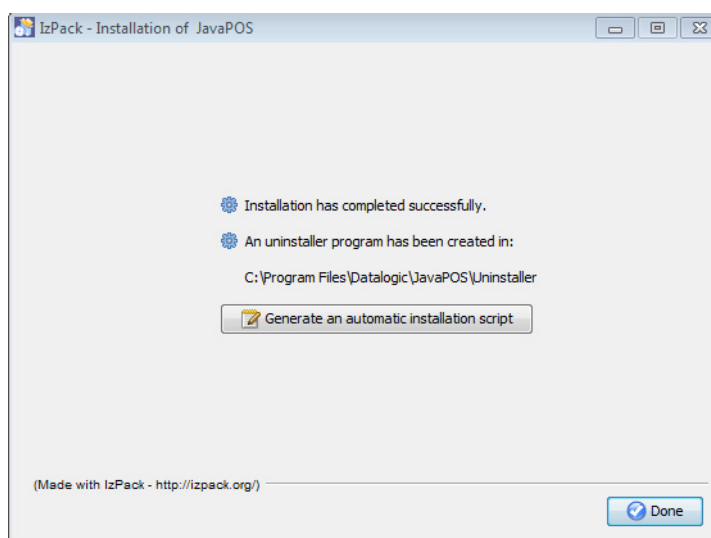
5. Click **Next**. An installation progress dialog is displayed.



6. If applicable, click **Next** for updating Java environment variables.



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



Linux Installation

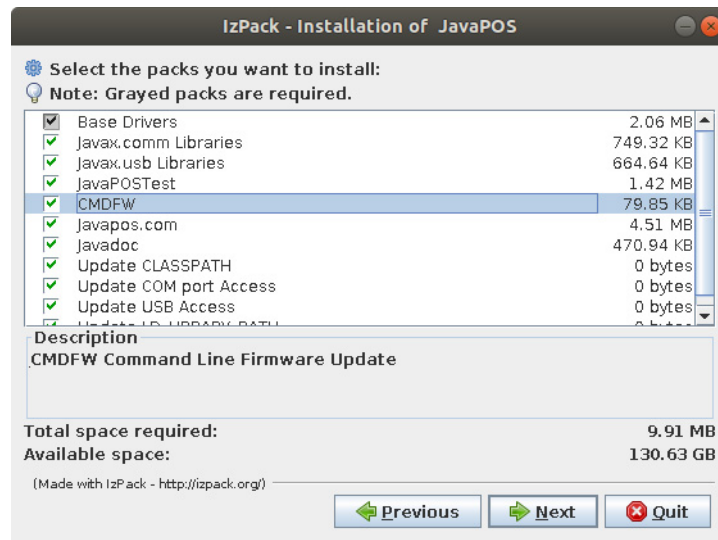
As with the Windows operating system, Linux requires version 1.7 or later of the Java Runtime Environment. Visit <http://www.oracle.com/technetwork/java/index.html> for download and installation instructions. The Java application must exist in your system path in order for the Datalogic JavaPOS application links to function properly.

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 `JavaPOS_Setup.jar` file exists (remember, Java must be found in the system path):

```
sudo java -jar JavaPOS_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:



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 JavaPOSTest 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.

The install directory contains a file that can be used to launch JavaPOSTest.

- For Windows: `JavaPOSTest.bat`
- For Linux: `JavaPOSTest.sh`

On Linux, you can run JavaPOSTest from a command prompt or terminal window by issuing the following command:

```
sudo ./JavaPOSTest.sh
```



NOTE

JavaPOSTest should always be run with sudo privileges. Shell script may also need the executable file permission added to it by running
`sudo chmod +x JavaPOSTest.sh`

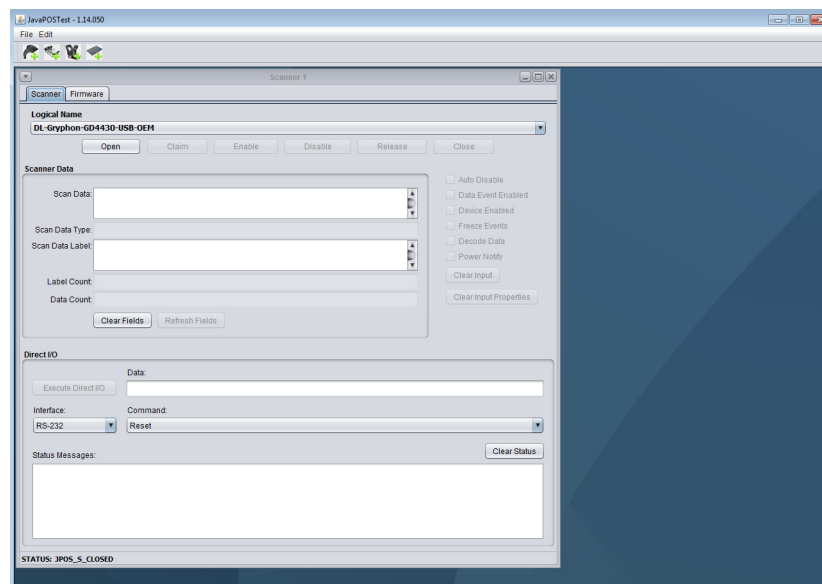
`JavaPOSTest.sh` (`.csh`) is located in the JavaPOS directory (depending upon which shell is used).



NOTE

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

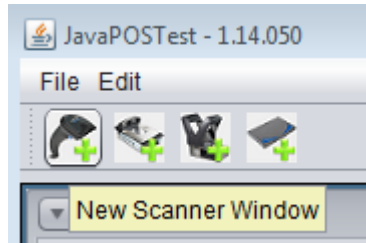
After JavaPOSTest starts, the following screen is displayed:



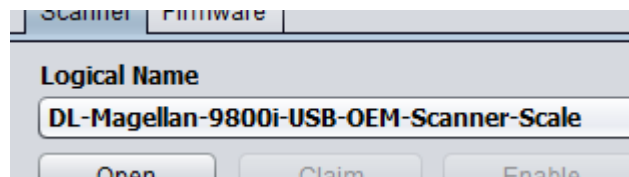
Example - Configuring a Scanner

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. If no scanner windows are currently open, click the **Scanner** button.



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



4. Click **Open**, and then **Claim**.
5. Click the **Enable** button, or manually select the **Data event enabled**, **Device enabled**, and **Decode data** checkboxes.
6. Scan an item. If the scanner is properly connected and configured, the scan data is automatically displayed.

If the Logical Name field is empty or does not contain the profile name, JavaPOSTest was unable to locate the proper `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.

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 1.14 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.jar
./dls.properties
./brand.properties
./SupportJars/commons-lang3-3.8.1.jar
./LabelIdentifiers.csv
./SupportJars/log4j2.xml
./SupportJars/log4j-api-2.11.1.jar
./SupportJars//log4j-core-2.11.1.jar
./jpos.xml
./ECIEncoding.csv
./SupportJars/DLRFIDLibrary.jar
```

“javax.comm. Libraries”

```
./SupportJars/jssc.jar
./SupportJars/nrjavaserial-3.15.0.jar
./SupportJars/libLinuxSerialParallel.so (Linux ONLY)
./SupportJars/libLinuxSerialParallel_g.so (Linux ONLY)
```

“javax.usb Libraries”

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

"JavaPOSTest"

```
./SupportJars/JavaPOSTest.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
./JavaPOSTest.bat (Windows ONLY)
./JavaPOSTest.sh (Linux ONLY)
```

"CMDFW"

```
./SupportJars/CMDFW.jar
./SupportJars/jargs.jar
./cmdfw.properties
./cmdfwc.bat (Windows ONLY)
./cmdfwc.sh (Linux ONLY)
./cmdfwc.csh (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)
./cmdfw.sh (Linux ONLY)
```

"javapos.com"

```
./SupportJars/jcl.jar
./SupportJars/jpos.properties
./SupportJars/jposl14-controls.jar
./SupportJars/jposl14.jar
./SupportJars/jpos-dls-ext.jar
./SupportJars/xercesImpl-2.12.0.jar
./SupportJars/xml-apis-2.12.0.jar
./SupportJars/jna-5.2.0.jar
./SupportJars/jna-platform-5.2.0.jar
```

“Update CLASSPATH”

Windows

A new system environment variable `JAVAPOS_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\appframework-
1.0.3.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\beansbinding-
1.2.1.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\CMDFW.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\commons-lang3-
3.8.1.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\DLRFIDLibrary.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jargs.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\JavaPOSTest.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jcl.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jna-5.2.0.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jna-platform-
5.2.0.jar;C:\Program Files\Datalogic\JavaPOS\SupportJars\jpos-
dls-ext.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jpos114-
controls.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jpos114.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jsr80.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jsr80_r.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\jssc.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\log4j-api-
2.11.1.jar;C:\Program Files\Datalogic\JavaPOS\SupportJars\log4j-
core-2.11.1.jar;C:\Program
Files\Datalogic\JavaPOS\SupportJars\nrjavaserial-
3.15.0.jar;C:\Program Files\Datalogic\JavaPOS\SupportJars\swing-
layout-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\xercesImpl-
2.12.0.jar;C:\Program Files\Datalogic\JavaPOS\SupportJars\xml-
apis-2.12.0.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=/usr/local/Datalogic/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=<install_path>/JavaPOS/SupportJars`

to ensure proper libraries are used

PATH — WINDOWS ONLY

During the installation process a new system environment variable by the name of JAVAPOS_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.

Uninstall

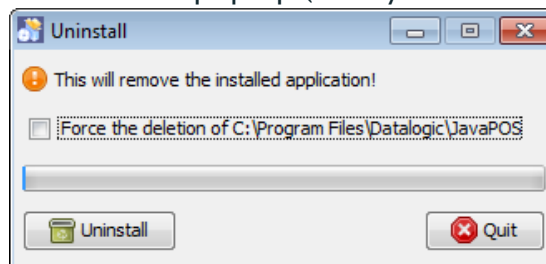
To uninstall Datalogic JavaPOS, follow these steps:



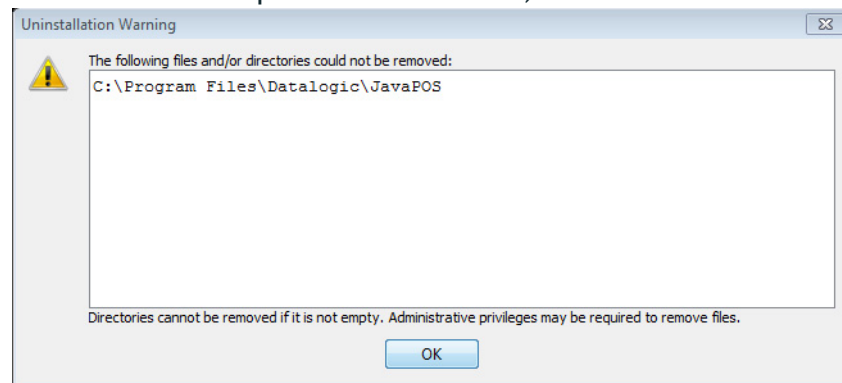
NOTE

Instructions may differ slightly depending on the current version of Datalogic JavaPOS. Please contact your support representative if you have any questions or are making a significant upgrade in JavaPOS version.

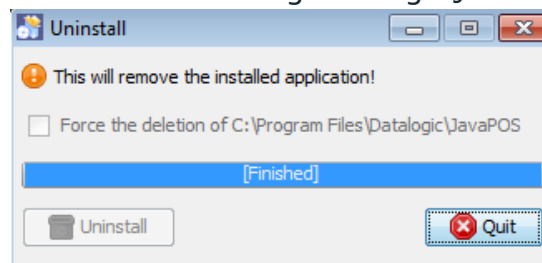
1. Locate the `uninstaller.jar` file in the `<install_directory>/JavaPOS/Uninstaller/` folder
2. Double click on the file (**Windows only**), or run the following command in a terminal window: `java -jar uninstaller.jar`
3. An uninstaller window will pop up (it may take a few moments):



- Click **Uninstall**. A few files may remain due to permissions (you may need to run as administrator outside of the JavaPOS folder and choose the forced deletion option to remove all)



- Press quit and manually delete any additional files that may remain (delete folders also if not installing Datalogic JavaPOS again).



Device Service Control

The Datalogic Device Controls use two property files to control certain optional operations. These files are named `dls.properties` and `brand.properties`. Upon initial service startup, the service will implement the following search hierarchy for the files using the first instance it finds:

- Current working directory of application launch (./)
- Java Classpath
- System configured temp directory

If the service is unable to locate either of the files, it will operate using the default values for each option.

dls.properties file

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

Property: `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: `AvalancheEnabled`

Default Value: `true`

Description: If set to true, a `.prf` file is created for use with Avalanche.

Property: `BtWaitTime`

Default Value: `1000`

Description: Denotes the number of milliseconds to wait for a Bluetooth device to enumerate on the port.

Property: `ConvertBCDtoASCII`

Default Value: `true`

Description: When true the data from the scanner is converted to ASCII for numbers 0 thru 9.

Property: `ECIEncodingFile`

Default Value: `./ECIEncoding.csv`

Description: Denotes the path to where the ECI encoding file is located. Path may be relative or absolute.

Property: `EnableDisablePoll`

Default Value: `false`

Description: If set to true, the current state of enabled or disabled will be continuously sent to the scanner at regular intervals.

*This property was created for a few legacy models. It should only be used under certain circumstances.

Property: `EnableDisablePollRate`

Default Value: `1`

Description: Denotes the number of seconds between polling operations.

Property: `EnableECI`

Default Value: `false`

Description: If set to true, Extended Channel Interpretation (ECI) will be used for data labels.

Property: `FirmwareSendReset`

Default Value: `true`

Description: If set to true, a reset command will be sent after Host Down-load.

Property: `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: `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: ImageDest

Default Value: ./

Description: Denotes the folder to store captured images.

Property: LabelIdentifiersFile

Default Value: ./LabelIdentifiers.csv

Description: Denotes the path to the label identifiers file. Path may be relative or absolute.

Property: ObjWaitTime

Default Value: 100

Description: Denotes the number of milliseconds to wait for IHS and beep command on USB-OEM scanners.

Property: PollRateForPortChange

Default Value: 1000

Description: Denotes the rate in milliseconds to poll the port for USB-COM changes on Linux.

Property: PostRemovalErrorEvents

Default Value: false

Description: If set to true, error events will still be enqueued after device removal.

Property: ResetTimeout

Default Value: 60000

Description: Denotes the number of milliseconds to wait for a reset to finish before throwing an error.

Property: RetryWaitTime

Default Value: 1000

Description: Denotes the number of milliseconds to wait before issuing a retry for certain USB-OEM tasks.

Property: RFIDMaxRetry

Default Value: 10

Description: Denotes the maximum number of retries to execute before failing a read or write operation on an RFID tag.

Property: 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: SendCookedData

Default Value: false

Description: If set to true, the data that is in the scanDataLabel property is copied into the scanData property.

Property: SuppressErrors

Default Value: false

Description: If set to true, error events will not be delivered to the host.

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, MOF files, etc.

Property: 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: Timeout

Default Value: 5000

Description: Denotes the number of milliseconds to wait for a response to most commands before throwing an error.

Property: UseClaimLockFile

Default Value: false

Description: If set to true, a lock file will be generated when a device is claimed.

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 Chapter 34 of 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 and USB-OEM 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:



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

- | | |
|--------------------|------------------|
| • Motor | • Internal Scale |
| • Horizontal Laser | • Remote Display |
| • Vertical Laser | • 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.

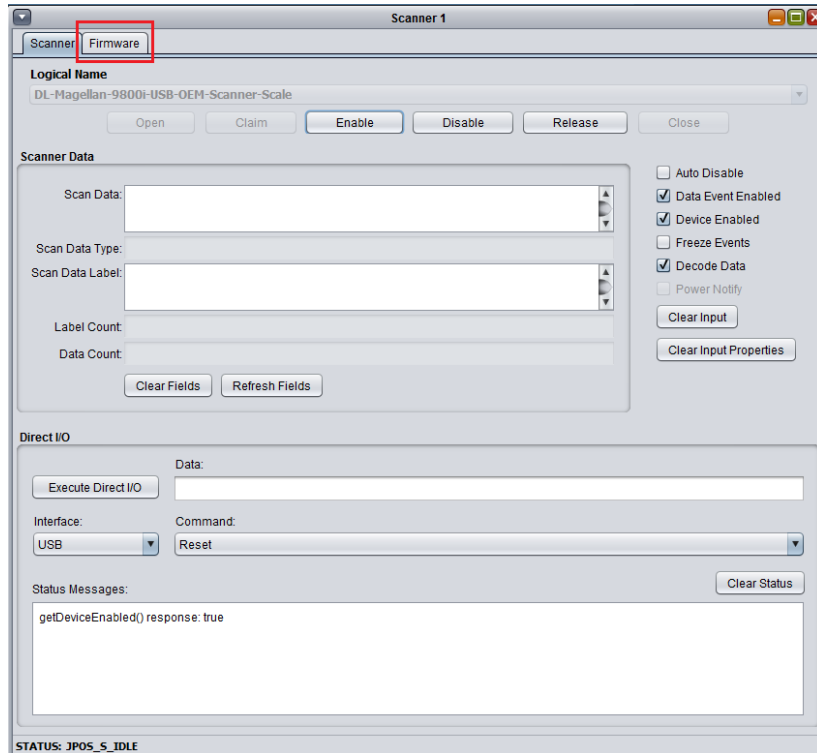
Supported statistics are:

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

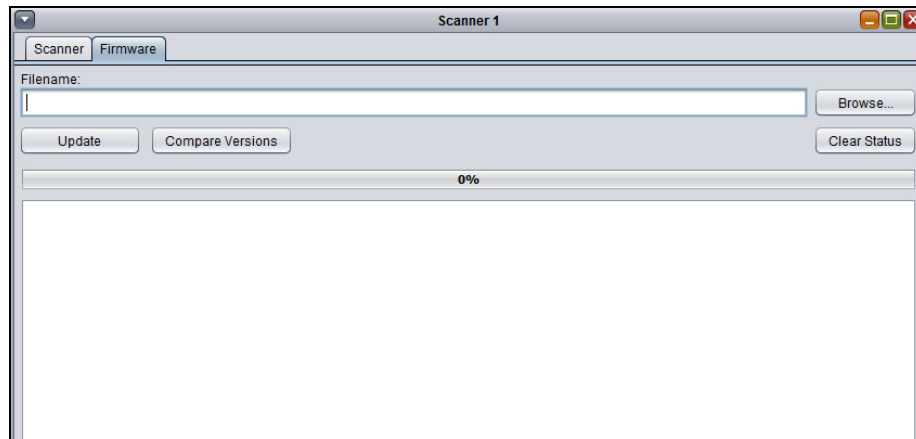
CapUpdateStatistics = False

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

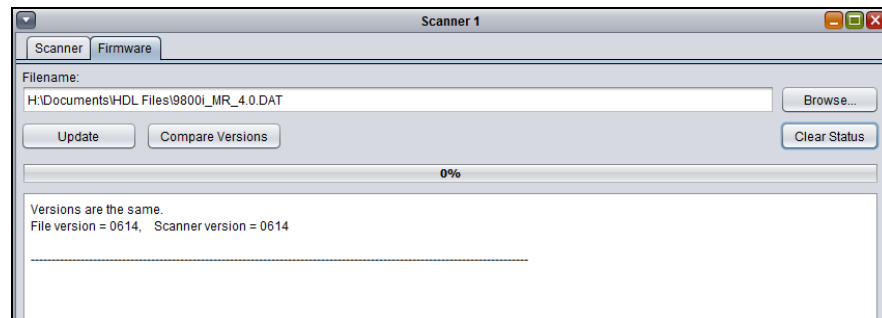
Firmware Update



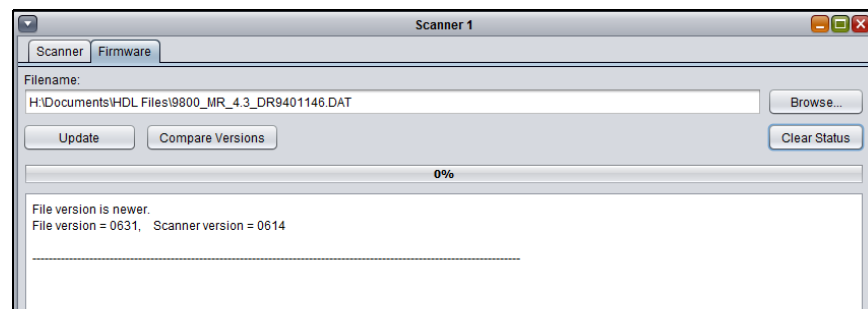
Upon selecting the “Firmware” update button, the following dialog will be presented. The user can then browse to the required update firmware file. There are two separate firmware file types: USB or RS-232. Each has the .DAT or .S37 extension, respectively.



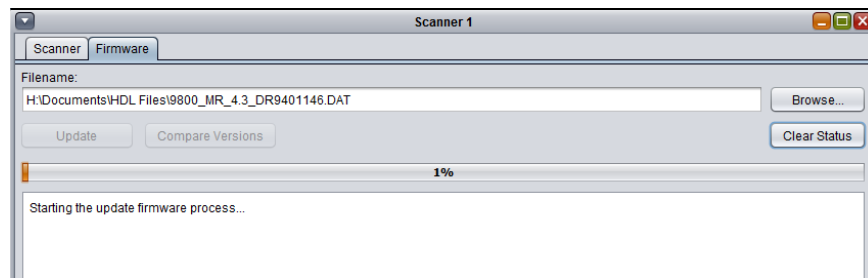
The dialog box will let the user compare or update the firmware. The following is a screenshot of a successful compare when the versions are equal.



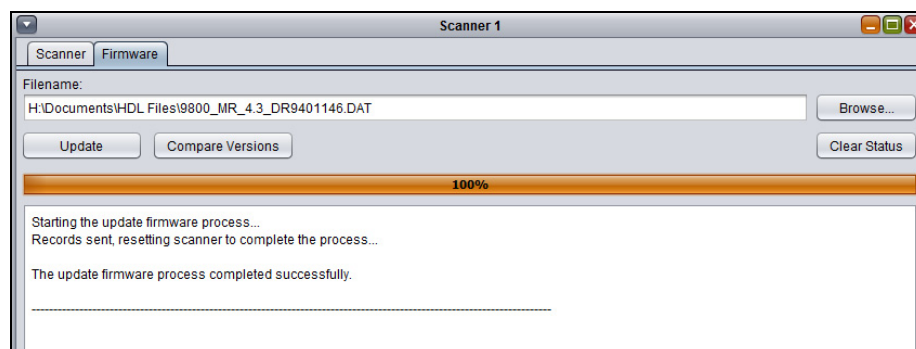
The following is a screenshot of a successful compare when the versions are different.



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



Below is an example of a successfully completed firmware update.



Standard JavaPOS Scale operations

See Chapter 33 of 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"`.

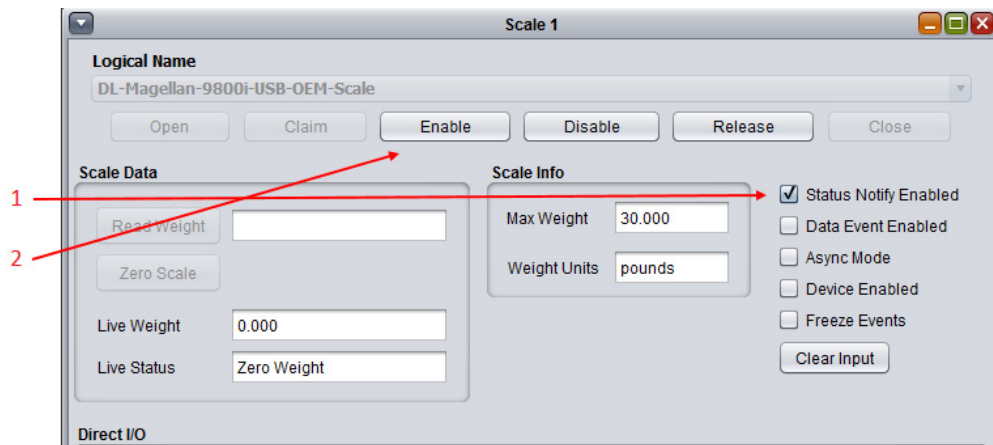
Display

Scale supports a remote display, but does not support text display. The POS application cannot send data to the scale's remote display (`CapDisplay-Text = 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.”



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.

Standard JavaPOS RFID operations

See Chapter 32 of the *Unified POS Retail Peripheral Architecture v1.14* document for Properties and Methods. You may also find a simplified implementation example in the `<install_directory>/JavaPOS/example/com/jpos/example/` folder.

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
Device Information	3001	None	Output: 2 byte Vendor ID (big-endian)
Request Scanner Generic Management Information	3002	None	Output: variable length ASCII bytes
Request Scanner Vendor-specific Management Information	3003	None	Output: variable length ASCII bytes
Request Scale Generic Management Information	3004	None	Output: variable length ASCII bytes
Request Scale Vendor-specific Management Information	3005	None	Output: variable length ASCII bytes

Scanner Command	Cmd	int[] data	Object object
Request Read of the Digital Watermark Label Filter	3006	None	Output: variable length ASCII bytes
Request Clear of the Digital Watermark Label Filter	3007	None	Output: status string
Request Read of the Digital Watermark Label Filter	3008	Input: variable length bytes	Output: status 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.

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



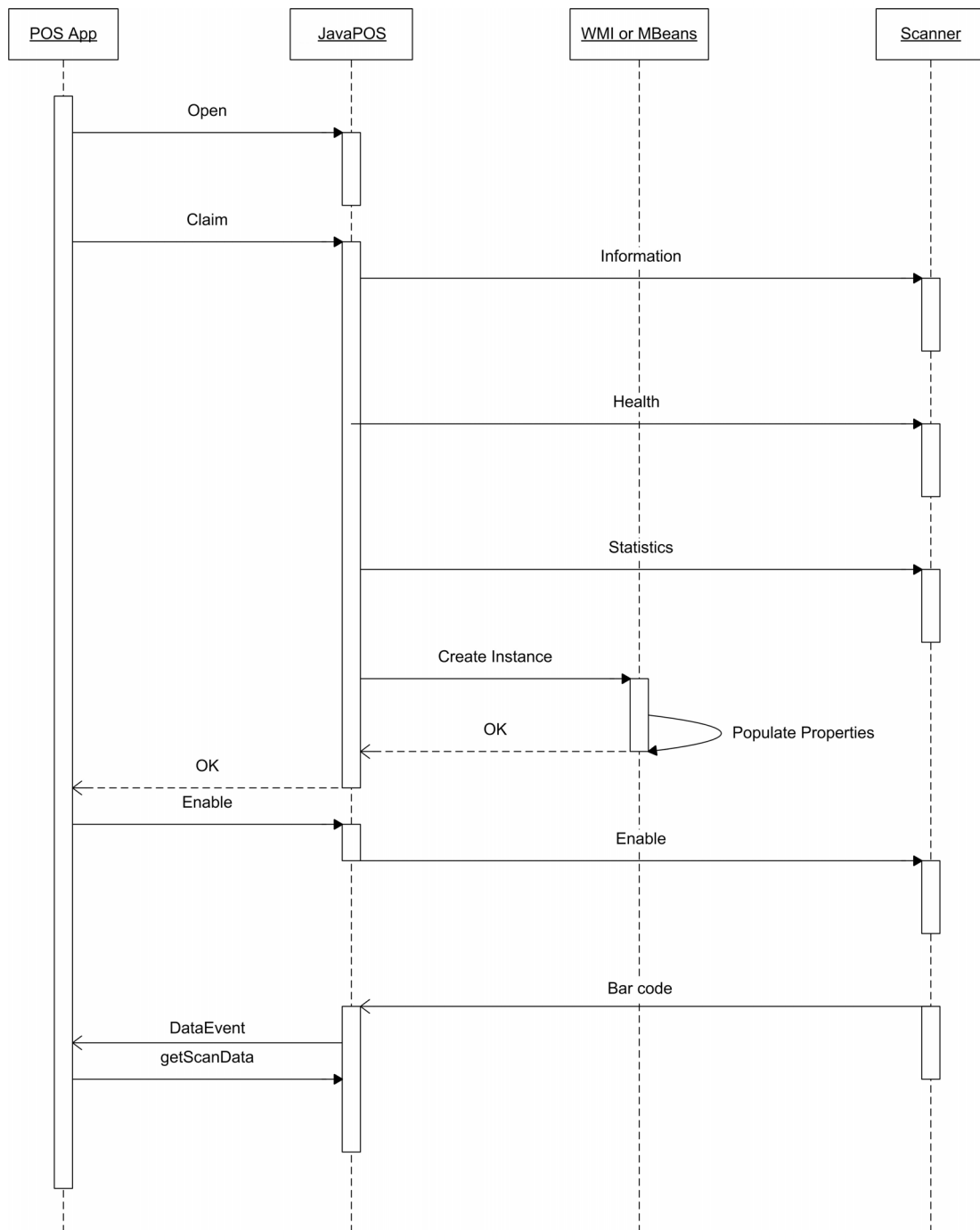
Appendix A. Remote Management Support

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.

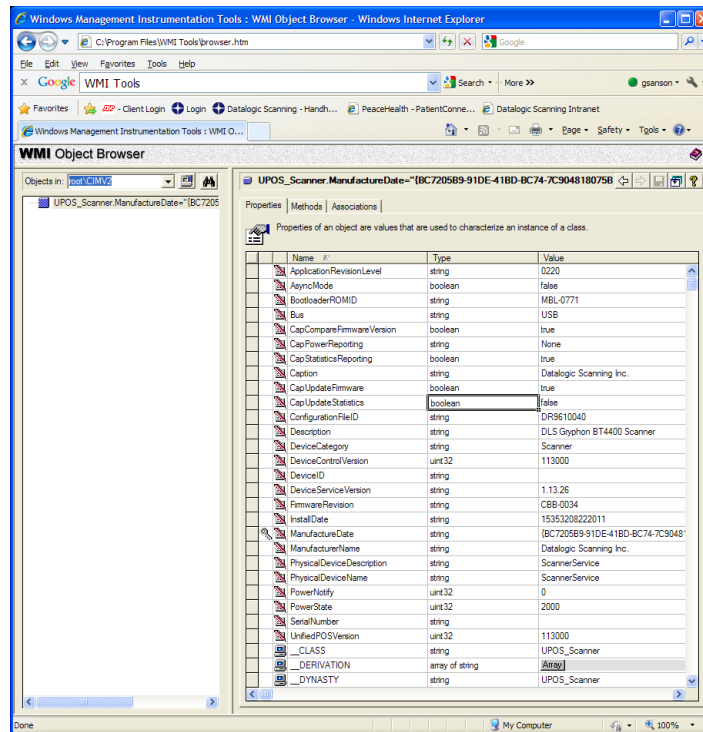
The current methods for statistical data retrieval use either the WMI (Windows Management Interface) or the Java JMX (Java Management Extension) using Managed Beans or MBeans. Each interface uses the `root/CIMV2/UPOS_Scanner` or `root/CIMV2/UPOS_Scale` structure. The following screenshots show the WMI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



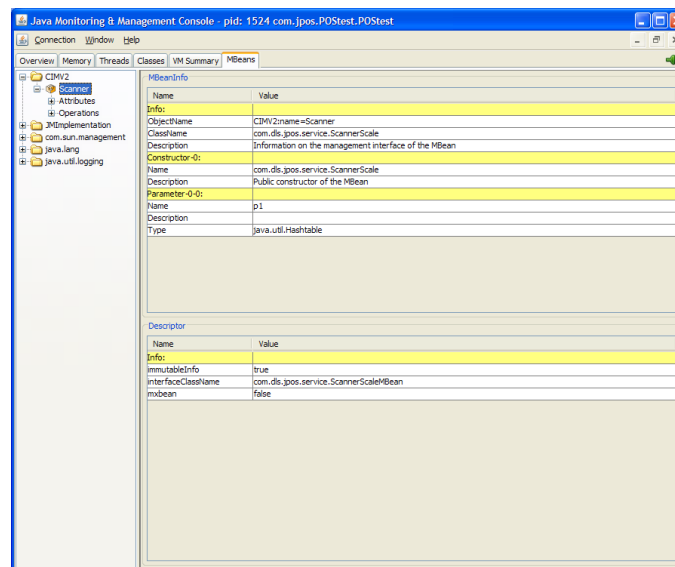
JMX MBeans Browser using jconsole

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

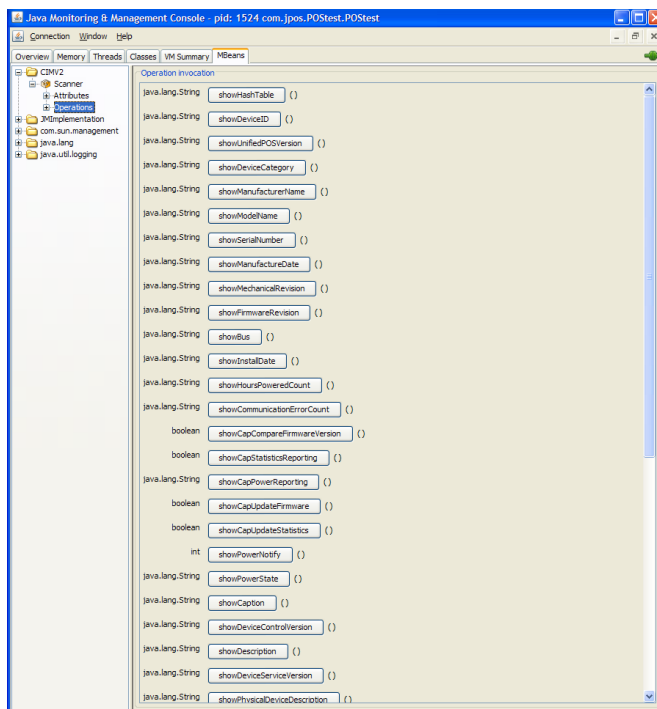
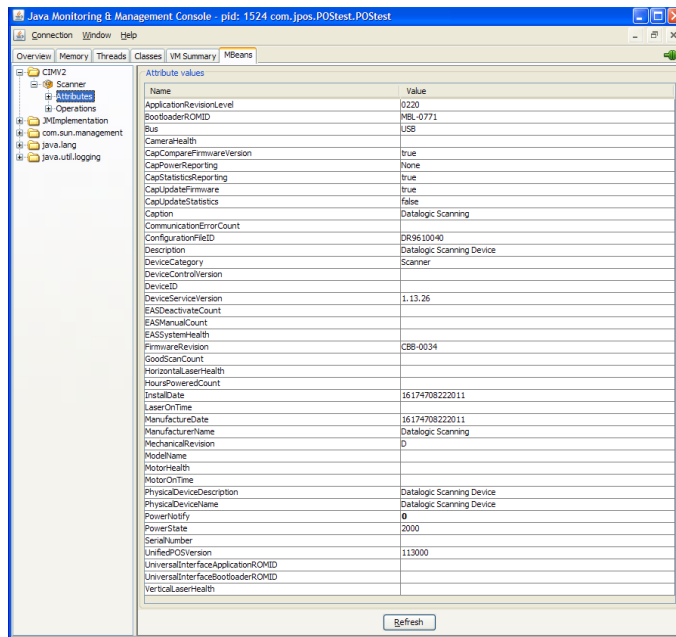
```
-Dcom.sun.management.jmxremote
```

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

JMX MBeans Browser using jconsole



JMX MBeans Browser using jconsole (continued)



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
#
```

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 dls.properties file, dls.properties, and property descriptions.

Property: AutoLoadConfig

Default Value: true

Property: AvalancheEnabled

Default Value: true

Property: BtWaitTime

Default Value: 1000

Property: ConvertBCDtoASCII

Default Value: true

Property: ECIEncodingFile

Default Value: ./ECIEncoding.csv

Property: EnableDisablePoll

Default Value: false

Property: EnableDisablePollRate

Default Value: 1

Property: EnableECI

Default Value: false

Property: FirmwareSendReset

Default Value: true

Property: GenerateStatsOnEveryClaim

Default Value: true

Property: HDLRecordRetry

Default Value: 3

Property: ImageDest

Default Value: ./

Property: LabelIdentifiersFile

Default Value: ./LabelIdentifiers.csv

Property: ObjWaitTime
Default Value: 100

Property: PollRateForPortChange
Default Value: 1000

Property: PostRemovalErrorEvents
Default Value: false

Property: ResetTimeout
Default Value: 60000

Property: RetryWaitTime
Default Value: 1000

Property: RFIDMaxRetry
Default Value: 10

Property: ScannerInfoFilename
Default Value: info

Property: SendCookedData
Default Value: false

Property: SuppressErrors
Default Value: false

Property: TempDir
Default Value: %Temp% (Windows), /tmp (Linux)

Property: ThrowExceptionOnScaleMotion
Default Value: false

Property: Timeout
Default Value: 5000

Property: UseClaimLockFile
Default Value: false

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: bluetoothDelayStatistics
Default Value: False
Description: Used to delay statistics gathering for Bluetooth devices. Time determined by the value of the BtWaitTime property.

Property: bluetoothScannerAlwaysEnabled
Default Value: False
Description: Used to keep Bluetooth devices in the enabled state without sending any commands.

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.

Property: `deviceClass`

Default Value: RS232Scanner

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

Property: `imageBuffers`

Default Value: 1

Description: number of buffers the connected application maintains

Property: `ipAddress`

Default Value: 192.168.0.2

Description: Address number used to communicate with the PortalScanner

Property: `ipPort`

Default Value: 26666

Description: Port number used to communicate with the PortalScanner

Property: `laneNumber`

Default Value: 0

Description: lane number used to identify location of 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: `serviceVersion`

Default Value: (current service version number)

Description: used for messages to and from PortalScanner

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: `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: `WMIEnabled`

Default Value: False

Description: Enables WMI data store population

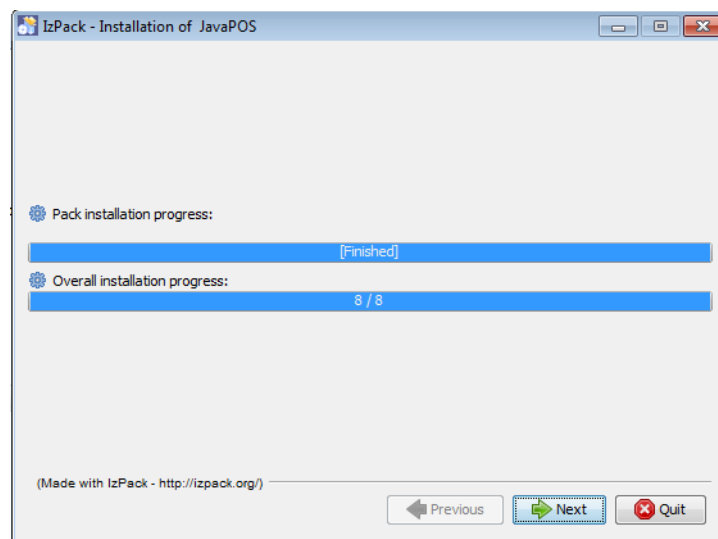


Appendix B. Auto Installation Script

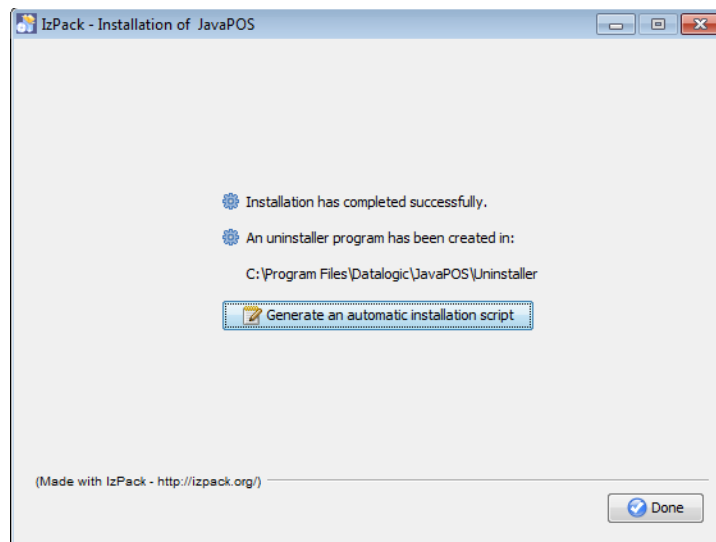
Creation and use of auto-installation script

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

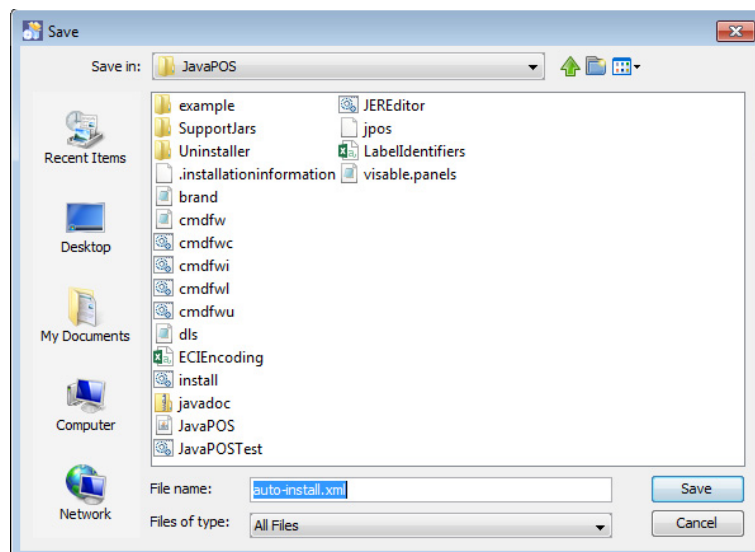
Figure 10. After normal installation is finished



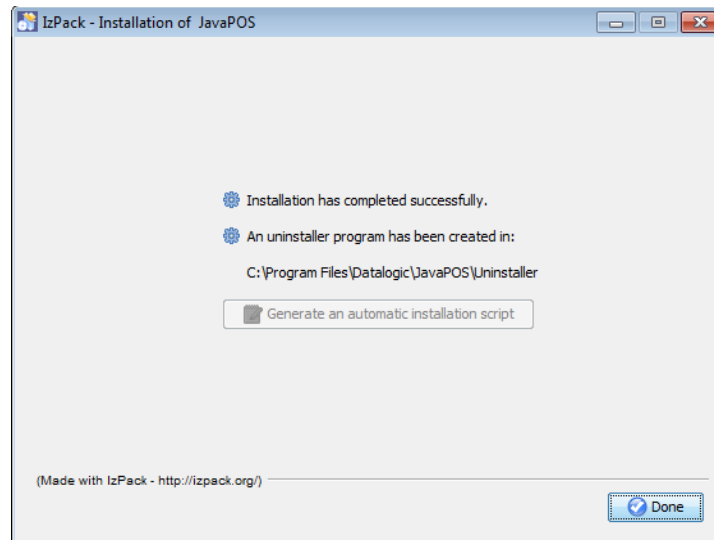
1. Click the "Generate an automatic installation script" button



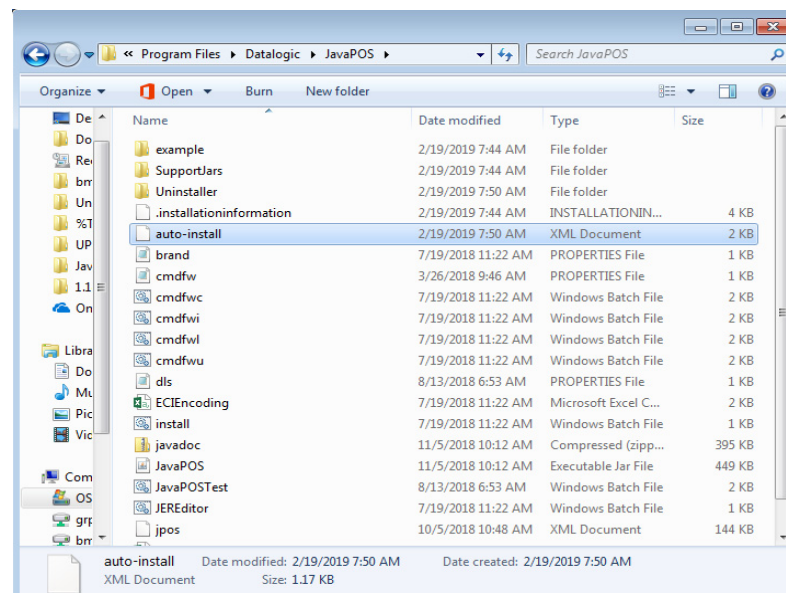
2. A "Save As" dialog pops up. Click save ("auto-install.xml" is the default file name).



3. The file has now been created.



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



Use this file along with `JavaPOS_Setup.jar` to perform command line installation.

Command Line Installation

```
java -jar JavaPOS_Setup.jar auto-install.xml
```

Contents of auto-install.xml file (example)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AutomatedInstallation langpack="eng">
<com.izforge.izpack.panels.hello.HelloPanel
id="HelloPanel_0"/>
<com.izforge.izpack.panels.licence.LicencePanel
id="LicencePanel_1"/>
<com.izforge.izpack.panels.target.TargetPanel
id="TargetPanel_2">
<installpath>C:\Program Files\Datalogic\JavaPOS</
installpath>
</com.izforge.izpack.panels.target.TargetPanel>
<com.izforge.izpack.panels.packs.PacksPanel
id="PacksPanel_3">
<pack index="0" name="Base Drivers" selected="true"/>
<pack index="1" name="Javax.comm Libraries"
selected="true"/>
<pack index="2" name="Javax.usb Libraries"
selected="true"/>
<pack index="3" name="JavaPOSTest" selected="true"/>
<pack index="4" name="CMDFW" selected="true"/>
<pack index="5" name="Javapos.com" selected="true"/>
<pack index="6" name="Javadoc" selected="true"/>
<pack index="7" name="Update CLASSPATH" selected="true"/
>
</com.izforge.izpack.panels.packs.PacksPanel>
<com.izforge.izpack.panels.install.InstallPanel
id="InstallPanel_4"/>
<com.izforge.izpack.panels.process.ProcessPanel
id="ProcessPanel_5"/>
<com.izforge.izpack.panels.finish.FinishPanel
id="FinishPanel_6"/>
</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  
jna-5.2.0.jar  
jna-platform-5.2.0.jar  
jpos114.jar  
jpos114-controls.jar  
jpos-dls-ext.jar  
jsr80.jar  
jsr80_ri.jar  
jsr80_linux.jar(Linux)  
nrjavaserial-3.15.0.jar  
xercesImpl-2.12.0.jar  
xml-apis-2.12.0.jar  
cmdfw.jar  
log4j-api-2.11.1.jar  
log4j-core-2.11.1.jar
```

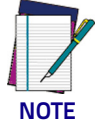
Then the Java executable can be invoked with the following command line options:

Windows (cmdfwu.bat):

```
java -cp ".\*;.\SupportJars\*" -  
Djava.ext.dirs=.\;.\SupportJars -  
Djava.library.path=.\;.\SupportJars  
cmdfw.Cmdfw -s DL-Magellan-USB-COM-Scanner -f  
DR9400414.S37 -v
```

Linux (cmdfwu.sh):

```
java -cp "/*:../SupportJars/*"
-Djava.ext.dirs=../SupportJars
-Djava.library.path=../SupportJars cmdfw.Cmdfw
-s DL-Magellan-USB-COM-Scanner -f
DR9400414.S37 -v
```

**NOTE**

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

Help output**cmdfw version: 1.14.050**

```
Usage: cmdfw.Cmdfw [-h, --help] [-i, --info] [-l, --list] [-c,
--compare] [-s, --scanner] <ScannerName> [-f, --filename]
<Filename> [-b, --startBaud] <StartingBaud> [-t, --endBaud]
<EndingBaud> [-u, --force] [-e, --enabled] [-v, --verbose]
```

The default behavior is to update the firmware using the [-s, --scanner] <ScannerName> [-f, --filename] <Filename> options

Options

-h, --help	print usage information
-i, --info	save info to file (default: <DeviceClass>_info.txt*), -s, --scanner<ScannerName> is required
-l, --list	list all available ScannerNames in jpos.xml file
-c, --compare	compare <FileName> with firmware in <ScannerName>, --scanner <ScannerName> & --filename <FileName> are required
-s, --scanner	scanner target (logical name from jpos.xml)
-f, --filename	file target for updating/comparing
-b, --startBaud	set the starting baud rate for serial devices
-t, --endBaud	set the ending baud rate for serial devices
-u, --force	force update regardless of file version passed in
-e, --enabled	leave scanner enabled after update/compare/info command
-v, --verbose	print operational information to the console
* <DeviceClass> refers to the DeviceClass property from jpos.xml	

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
```

NOTES

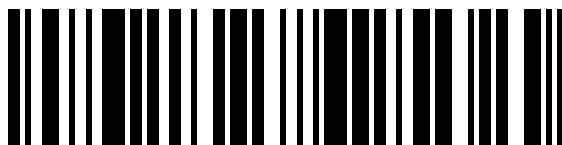
©2006 - 2019 Datalogic S.p.A. and/or its affiliates ▪ All rights reserved ▪ Without limiting the rights under copyright, no part of this documentation may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means, or for any purpose, without the express 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.



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. G)

May 2019