Installing the Mettler-Toledo JPOS Driver on **Linux** Systems

1. Setup your Java Environment:

In a Linux shell window enter the command: "*java -version*" to confirm that you have Java installed on the target Linux system.

1. Execute the "java -version" command.
    a. It should return something like: "java version 1.8.0_161".
2. If you don't have Java on your system, go to Google.com and type in "java download".
3. It is important for this JavaPOS that you install the correct 32 or 64 bit Java version.
    a. On 32-bit Linux OS's install a 32 bit Java.
    b. On 64-bit Linux OS's install a 64 bit Java.
    c. Notes:
        i. The Mettler Toledo JPOS Installer detects whether your Linux OS is 32 or 64 bits and installs the corresponding JPOS driver.
        ii. Mismatching your Java and OS (32 bit Java on a 64 bit Linux, for example) will not likely work with this JPOS driver and is not supported.
    d. To determine if your Linux OS is 32 or 64 bits:
        i. In the shell window execute the "arch" command.
            1. If "x86_64" is returned, then you have a 64-bit machine.
            2. If "x86" is returned, then you have a 32-bit machine.
        ii. "uname –a" is also a useful command.
    e. To determine if your Java is 32 or 64 bits:
        i. Execute the "*java -version*" command.

4. NOTE: *nrjavaserial.jar*: For Serial Communications, including USB Virtual COM Ports, the MT JavaPOS driver uses *nrjavaserial.jar* (and not *javax.comm.SerialPort*, *gnu.io.SerialPort*, *librxtxSerial.so,* etc.)

2. Installing JavaPOS:

1. To determine if you have an older *JavaPOS version* installed on your PC that needs to be removed, execute from the Linux shell:
    a. rpm –q –a | grep JavaPOS
        i. Returns, for example, *JavaPOS-1.8.20-3.x86_64*

2. To uninstall an older JavaPOS version, execute as super-user:
    a. *rpm –e JavaPOS-1.8.20-3.x86_64*   // for example.

3. Download the latest Linux JPOS installation package (.rpm) from the Retail Wiki site into a ***new folder*** on the target Windows machine.
    a. Note: the .rpm package should look something like this (it may be newer):
        i. JavaPOS-1.8.20-4.x86_64.rpm

4. Install JavaPOS by running the following RPM command in super-user mode:

rpm -ivh /home/~/*new folder*/JavaPOS-1.8.20-4.x86_64.rpm

5. This RPM command places the JavaPOS archived files in the following directory (unless the "/usr/local" prefix was changed in the RPM command line.):

   /usr/local/mt/javapos

6. REBOOT // this is required!!! Do it NOW!

3. Configuring JavaPOS:

1. Note the following directories/files in the */usr/local/mt/javapos*:
   a. 75-mt-scale-devices.rules
      i. This file recognizes an Ariva scale USB Virtual COM Port (if Ariva menu item 3.1 = 1) and assigns it to /dev/ttyS80. This rules file, which runs at Linux boot time, allows developers to assume that the Ariva Virtual COM Port will always be device /dev/ttyS80.
   b. JposLibs:
      i. Contains the Java archive files used by the driver and test application.
   c. MTScaleService:
      i. The MTScaleService/mtscale.jar file contains JavaPOS Device Service classes compliant with JavaPOS 1.82.
   d. nrjavaserialLib
      i. The nrjavaserialLib/nrjavaserial-3.9.3.jar file contains the communications drivers used by JavaPOS to communicate with the Ariva scale.
   e. jpos.xml:
      i. This is the configuration file used to configure your scale.
      ii. You'll need to configure this file, below, to match your POS system.
      iii. Contains three logical scale entries: Dialog06, Dialog06-PIPE and MettlerScale8217.
      iv. See below for editing instructions.
   f. jpos_orig.xml:
      i. Backup copy of the original jpos.xml.
   g. mtscaletestapp.jar
      i. The test application jar file.
   h. startMTScaleTestApptest.sh
      i. The shell script that runs the test application.
      ii. Contains CLASSPATH, LIB_DIR's, JAR_PATH's, etc.

2. Edit the */usr/local/mt/javapos/jpos.xml* configuration file as follows for the *logicalName=MettlerScale8217* or the *logicalName=MettlerScaleDialog06* entries.

   a. MAKE SURE YOU ARE EDITING THE CORRECT ENTRY!
      i. In N. American *logicalName=MettlerScale8217* will typically be used.
      ii. In Europe *logicalName=MettlerScaleDialog06* will typically be used.

   b. The *jpos.xml* file entry "<prop name="port" value="COM*n*"/>", the COM port name "COM1" needs to be changed to match whatever COM port you are using.
      i. For Linux Virtual COM Ports, typically /dev/ttyS80 will be setup when plugging in a Virtual COM Port device (such as an Ariva scale with menu setting 3.1 = 1).
         1. Note: /dev/ttyS80 originates from the *75-mt-scale-devices.rules* file.

c. The *jpos.xml* file entry <prop name="baudrate" value="9600"/> indicates the communication speed appropriate for the scale. If this property is missing or invalid then the default value 9600 is used.

d. Change the COM port parameters as needed (baud rate, parity, etc.).
i. Parity: 1 is for Odd Parity, 2 is for Even Parity, 0 is for No Parity.
ii. Stop bits: 0 or 1.
iii. Databits: 7 or 8.
iv. For the US and Canada scales are typically configured for 9600 baud, 7 data bits, Even parity and 1 stop bit, but this depends upon the POS system's configuration.
v. For Europe the Ariva scales are typically configured for 9600 baud, 7 data bits, Odd parity and 1 stop bit, but this depends upon the POS system's configuration.

e. The *jpos.xml* file entry “<prop name="metricUnit" value="false"/>” needs to be set to “true” or “false”.
i. If set to “false” the MeasuredWeight field will display weight in Lbs. or Oz.
1. This is typically true for the US.
ii. If set to “true” the MeasuredWeight field will display the weight in grams.
1. This is typically true for the EU.

f. The *jpos.xml* file entry "maximumWeight" also needs to be set as follows (based on the maximum capacity of the scale being used (see the scales' Data Label)):

i.	0-15 lbs.:	15000
ii.	0-6/6-15 lbs.:	15000
iii.	240 Oz.	240000
iv.	0-3/3-6 kg:	6000
v.	0-15 kg:	15000
vi.	0-6/6-15kg:	15000
vii.	0-30 lbs.	30000
viii.	0-15/15-30 lbs.	30000

g. The scale RS-232 settings must match the MTScale-JPOS RS-232 settings.
i. The Ariva RS-232 settings can be configured via the Ariva display menu.
ii. The Viva RS-232 settings can be configured via the Viva display menu.
iii. See the appropriate User Manuals to set these values.

h. Save the /usr/local/mt/javapos/*jpos.xml* file when done editing.

4. Running the Test Application:

1. Run the JavaPOS application within an XTerm window.
2. Super-user mode is not needed.
3. The *jpos.xml* file, above, must have been properly configured.
4. Change directory to /usr/local/mt/javapos.
5. From the command line, execute *./startMTScaleTestApptest.sh* to start the application.

a. Choose "*Select scale logical name*" based on the *jpos.xml* scale entry.
b. Select the protocol to be used.

    c. "Open scale" and "Claim scale".
    d. "Get weight" to validate scale is properly connected.

6. *The startMTScaleTestApptest.sh* command will set up the proper environment and start the JavaPOS test application. Depending on the configuration, other operations may be available as well such as Zero Scale, Tare Scale, etc.

5. Uninstalling the JavaPOS Application

1. To uninstall an older JavaPOS version, execute as super-user:
    a. *rpm –e JavaPOS-1.8.20-3.x86_64*   // for example.

2. This will remove all files under the */usr/local/mt/javapos* directory previously installed by the RPM command. Note**:** Some empty folders may remain.

6. Error Conditions:

1. Error: No weight data available
    a. Error message in log file:
        i. Error code = JPOS_E_TIMEOUT
        ii. Reported extended error code = SCALE_NODATA
        iii. Error message: No weight data available
    b. Possible cause:
        i. Ariva menu settings 3.1 – 3.6 do not match *jpos.xml* configuration settings.
        ii. Make sure you edited the correct "*JposEntry logicalName=*" entry in the *jpos.xml* file.

2. Test Application Permissions Error:
    a. As reported in the test application window:
        The device could not be claimed for exclusive access. Error code = JPOS_E_ILLEGAL
        >> Error message: invalid communication port: /dev/ttyS80
    b. As reported in XTerm window:
        i. RXTX fsh_lock() Error: opening lock file: /var/lock/LCK…ttyS80: Permission denied. FAILED TO OPEN: no such file or directory.
    c. Fixing this problem:
        i. This may only be a CentOS problem.
        ii. Note: a workaround is to run the application in super-user mode.
        iii. Also *sudo chmod 757 /var/run/lock*
        iv. But neither of these fixes are satisfactory.

7. Debugging:

1. Turn on the log file as follows:
    a. Set the *jpos.xml* file entry “<prop name="tracing" value="true"/>”.
    b. Set the *jpos.xml* file entry “<prop name="tracingLevel" value="DEBUG"/>”.

c. After running the test application, review the trace file for debugging information. Also send to Mettler Toledo for additional assistance.
d. *Three trace file are used: tracingScale_8217, tracingScale_COM1 and tracingScale_PIPE.*
e. The trace files are typically located here: */usr/local/mt/javapos*.