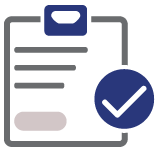
Terminology

The table below outlines the different course activities:

|  |  |
| --- | --- |
| ACTIVITY | DESCRIPTION |
| Demonstration | The Instructor will demonstrate the workflow, outlining the key concept(s). The student is not expected to replicate the Instructors demonstration; but understand the key concept(s) and workflow. |
| Lab | The Instructor will outline the key concepts, features and options. The student is expected to follow along with the instructor so that they understand the key concept(s), features and options for the Exercise |

The icon indicates an Info Tip. Info Tips help users understand unfamiliar workflows or actions.

The icon indicates that you need to be careful when implementing or configuring the step / option(s).

*The icon indicates Best Practice. A Best Practice is a method or technique that has been generally accepted as a standard way of doing things.*

Pre-requisites

The following pre-requisites need to be completed:

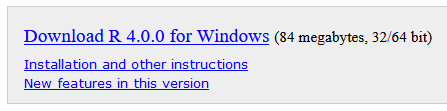
* Install R for Windows
* Set R Environmental Variables
* Install R Studio for Windows
* Configure Pentaho Data Integration with R

Installation of R

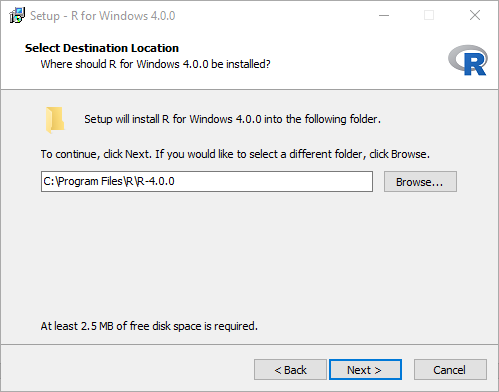
1. To install: download R from [r-project](https://cran.r-project.org/mirrors.html)

* Select a CRAN location (a mirror site) and click the corresponding link

1. Click on: the "Download R for Windows" link at the top of the page.
2. Click on: “install R for the first time" link at the top of the page.

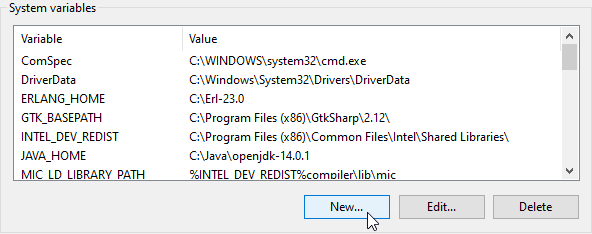


1. Click: Download R <version> for Windows
2. Run: R-4.0.0-win.exe file and follow the installation instructions.

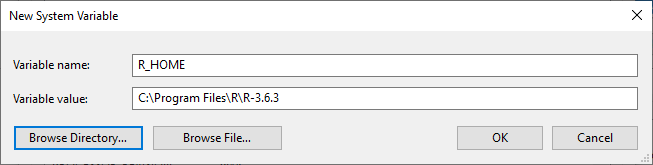


Set R Environmental Variables

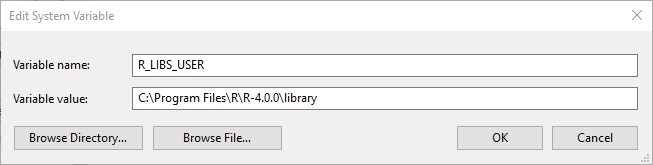
1. Go to: Control Panel > System > Advanced System Settings.
2. Click: Environment Variables button.
3. Under System Variables, click: New.



1. In the Variable Name field, enter: R\_HOME
2. Browse for the directory: C:\Program Files\R\R-3.6.3

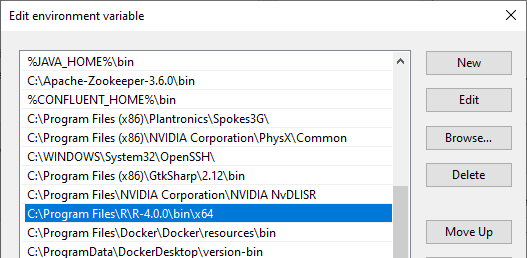


1. Repeat to add the variable: **R\_LIBS\_USER**
2. Browse for the directory: C:\Program Files\R\R-3.6.3\library



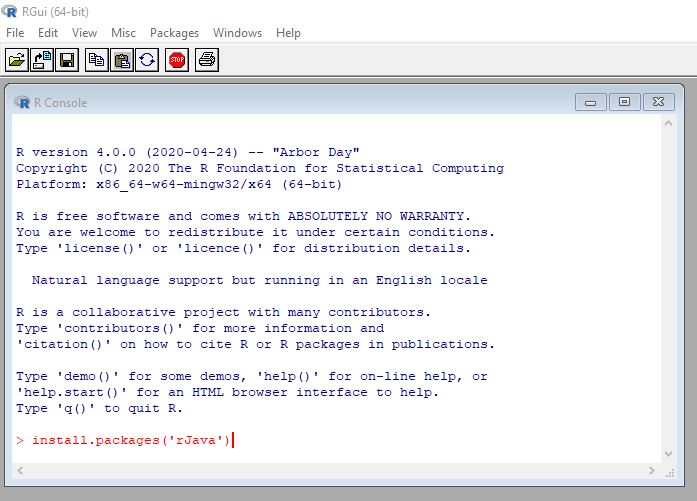
1. Add to the Path the location of the R executable: C:\Program Files\R\R-4.0.0\bin\x64

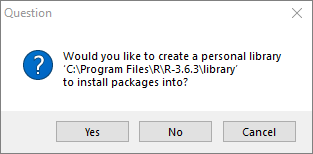
* Ensure the path references rcmd.exe and r.dll



1. Start R. In the R Console: Run the following command:

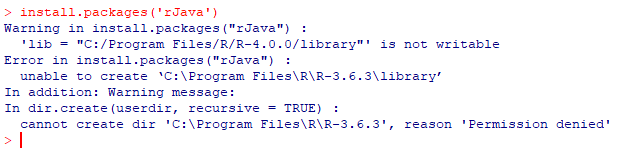
install.packages('rJava')

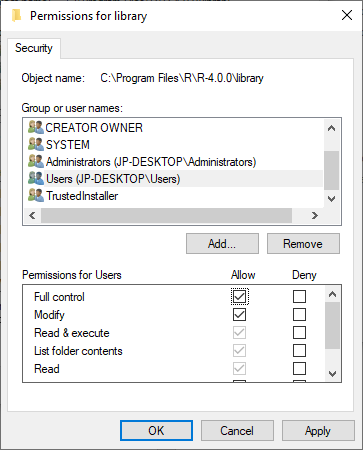


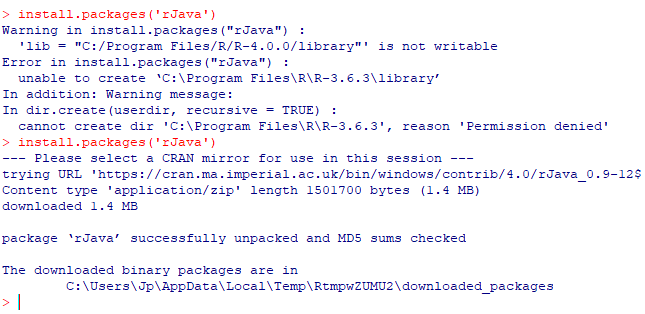
* If prompted with "Would you like to use a personal library instead?" click Yes.
* 
* If prompted with the path of the library, click Yes.
* When prompted for the CRAN mirror, choose a country then click OK.



*You may be denied permission writing to the library folder. You will need to change the permission for the folder.*



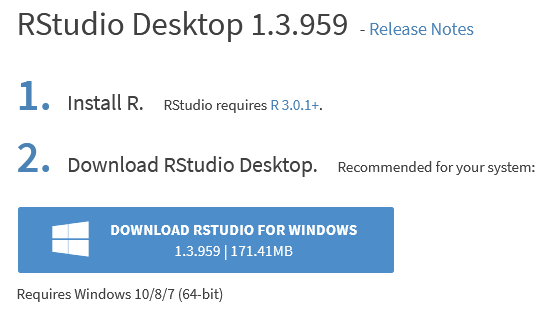




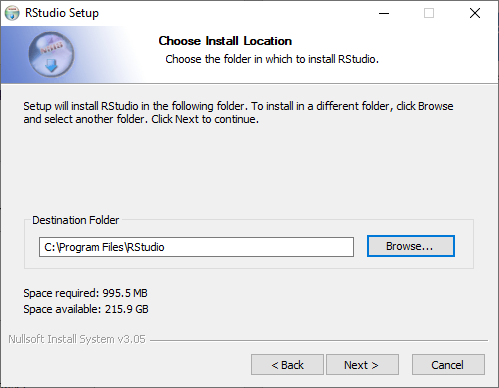
1. After rJava has successfully been installed, type q() to quit the R console.
2. Click Yes to close the workplace image.
3. Close R.

To Install RStudio

1. To install: download R Studio from [R Studio IDE](https://rstudio.com/products/rstudio/download/#download)
2. Click on the "Download RStudio for Windows" button.



1. Run: RStudio-1.3.959.exe file and follow the installation instructions.



Configure Pentaho Data Integration with R

In the rJava directory there is a jri.dll file that needs to be copied into the libswt directory of Spoon.

1. Stop: Spoon, if it's running
2. Find: %R\_LIBS\_USER%/rJava/jri/x64/jri.dll
3. Copy: jri.dll to the following directory

Windows: ﻿[Pentaho directory]/client-tools/data-integration/libswt/win64

Linux: [Pentaho directory]/client-tools/data-integration/libswt/linux



*Further details can be found at:* [R on PDI](https://support.pentaho.com/hc/en-us/articles/360000307943-Pentaho-Data-Integration#WPR)

Verifying Your Installation

1. Open a new transformation in PDI.
2. Drag an R Script Executor step onto the canvas.
3. Double-click the step and select the middle tab, R Script. You will see some comments at the top of the window:

# The main output is expected to be a data frame, unless "Output

# from script is text" is checked. So, to output a data frame the

# last statement in the script should be the name of the frame.

# In the case that the output is text (as would be seen on the

# R console), the last statement should be a "print" statement in

# order to print the object required.

1. Beneath the comment above, enter this code:

library(datasets)

iris

1. Once you have entered this code in the R Script tab, click the **Test Script** button on this tab.

# Apache Artemis Service

|  |  |  |
| --- | --- | --- |
| Introduction | In this guided demonstration, you will install Apache Artemis service. | |
| Objectives | In this guided demonstration, you will:   * Configure Apache Artemis Service |

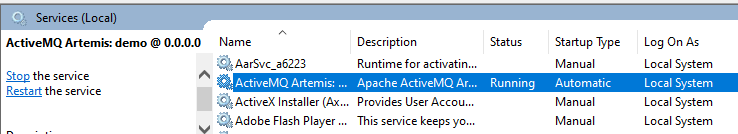
Step 1 - Artemis Service

On windows you will have the option to run ActiveMQ Artemis as a service.

1. CD to: C:\apache-artemis-2.13.0\bin\demo\bin>
2. Enter the following command to install it:

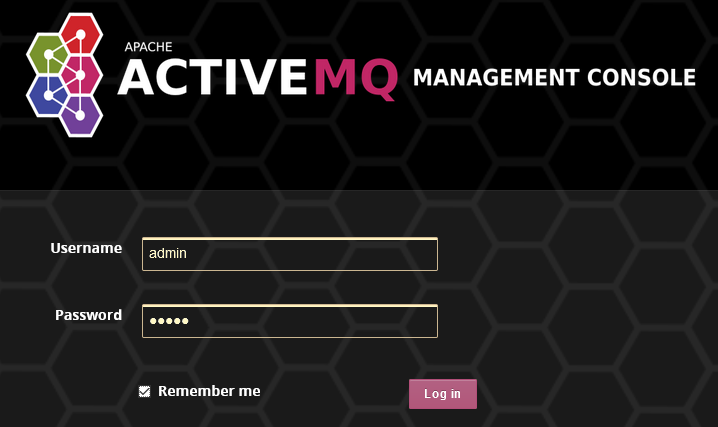
artemis-service.exe install

1. Start the Artemis Broker service.

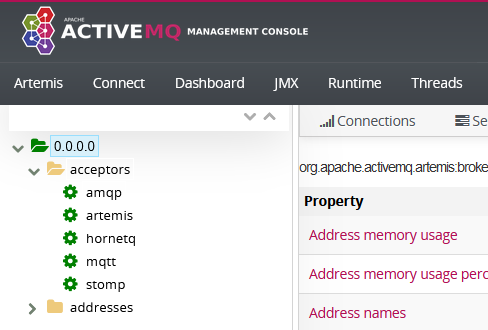


*Artemis listens on default port 61616 for all protocols*

1. In your browser, enter: localhost:8161

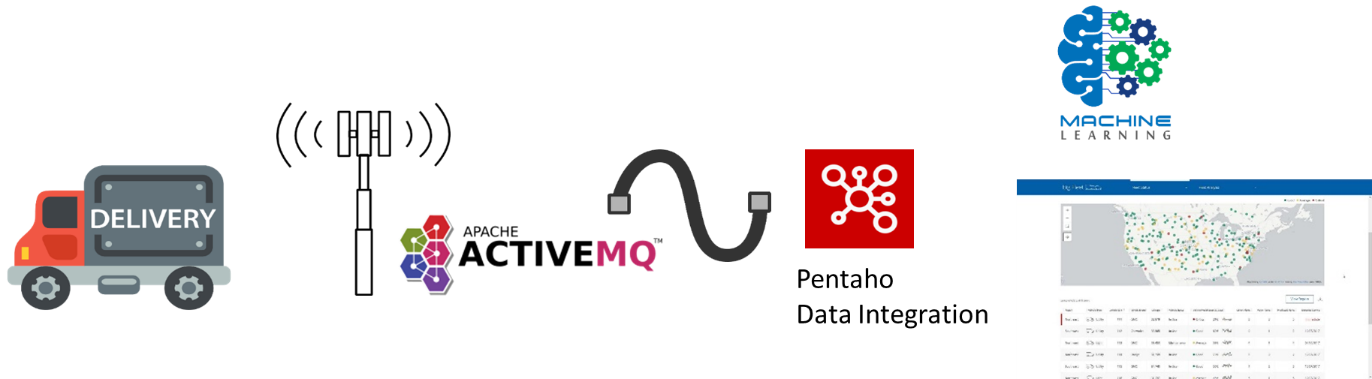


1. Expand acceptors



# Lab 2: Apache Artemis – IoT SensorData

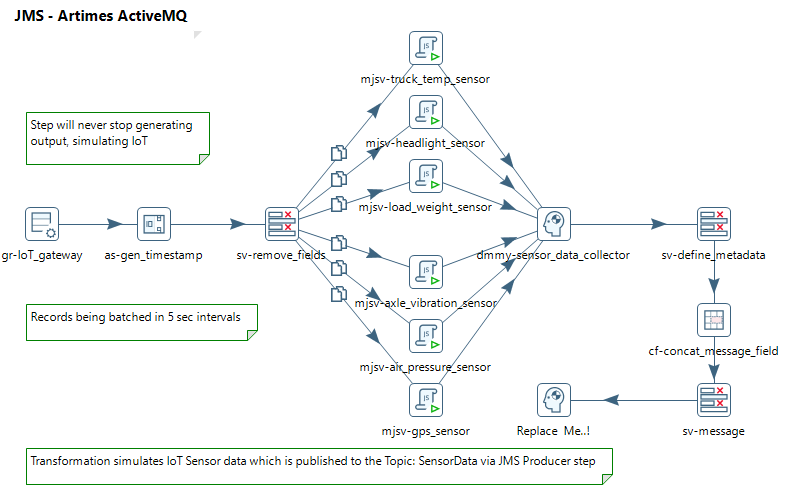
|  |  |  |
| --- | --- | --- |
| Introduction | A Logistics Company wish to track telemetrics from their fleet of delivery trucks. This will help with predictive maintenance and optimize delivery routes. | |
| Objectives | In this guided demonstration, the instructor will guide you through:   * Create a Queue: SensorData * Publish / Subscribe messages * Configure JMS Producer step * Configure JMS Consumer step * Configure Get records from stream step |



Step 1 - IoT Streaming Data (Transformation)

1. Open the Transformation:

C:\ Streaming--Data\02\_AMQP\Lab AMQP\tr\_mqtt\_producer.ktr



1. Replace the Dummy step with JMS Producer step.

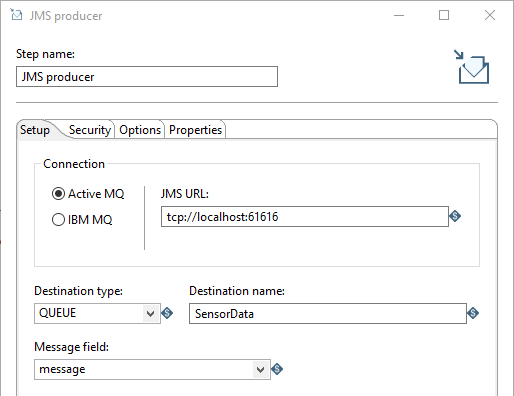
JMS Producer

The JMS Producer step ‘publishes’ the messages via a Queue: SensorData, to the Artemis ‘Demo’ Broker.

To configure the JMS Producer:

1. Drag and drop the JMS Producer step onto the canvas.
2. Double-click to set the properties as outlined below:

|  |  |  |
| --- | --- | --- |
| SETTING | DEFAULT VALUE | DESCRIPTION |
| Connection | Active MQ  tcp://localhost:61616 | JMS URL of Broker |
| Destination type | QUEUE | The Producer Topic |
| Destination name | SensorData | Specify the type pattern this exchange is using. |
| Message field | message | Input field from which this step will retrieve the message you are publishing. |



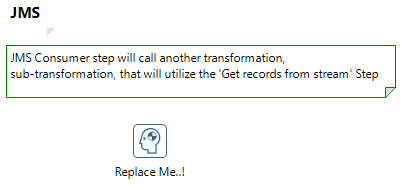
Do not RUN the Transformation until the JMS Consumer has been configured.

Step 2 - JMS Consumer

The JMS Consumer step batches the incoming stream.

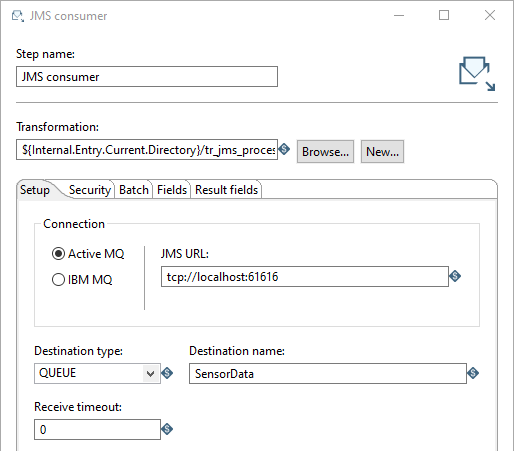
1. Open the Transformation:

C:\ Streaming--Data\03\_JMS\Lab Apache Artemis\tr\_jms\_consumer.ktr



To configure the JMS Consumer:

1. Drag and drop the JMS Consumer step onto the canvas.
2. Double-click to set the properties as outlined below:



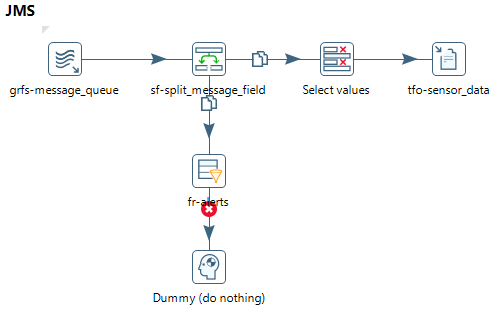
|  |  |  |
| --- | --- | --- |
| SETTING | DEFAULT VALUE | DESCRIPTION |
| Setup tab |  |  |
| Transformation | ${Internal.Entry.Current.Directory}  /tr\_jms\_process\_sensor\_data.ktr | Transformation that enriches the datastream |
| Connection | Active MQ  tcp://localhost:61616 | Specify the address of the Artemis Broker |
| Destination type | QUEUE |  |
| Destination name | SensorData | Specify the Artemis Queue name. |
| Message field | message | DIRECT routes messages to queues based on the message routing key. |
| Batch tab |  |  |
| Duration(ms) | 5000 | This value is the amount of time the step will spend collecting records prior to the execution of the transformation. |
| Number of records | 50 | After every ‘X’ number of records, the specified transformation will be executed, and these ‘X’ records will be passed to the transformation. |
| Max batch | 1 | Specify the maximum number of batches used to collect records at the same time. The default value is 1, which indicates a single batch is used for collecting records. |
| Fields |  |  |
| message | message | The incoming data stream message field |
| destination | destination | The name of the queue or topic from which the client consumes messages. |
| messageId | messageId | The value that uniquely identifies each message in a record |
| jmsTimestamp | jmsTimestamp | Specifies the time a message was generated |
| jmsRedelivered | jmsRedelivered | Specifies this message was marked for redelivery. This flag is useful if you need to de-duplicate messages |

Step 3 – Enrich the datastream

The datastream can be enriched via PDI.

1. Open the Transformation:

C:\ Streaming--Data\03\_JMS\Lab Apache Artemis\tr\_jms\_process\_sensor\_data.ktr



The message field is split into the various datastream fields which are appended to a Text File.

**

*As the file is being appended the data stream field names are not written to the file.*

Step 4 – Configure an Address and Queue

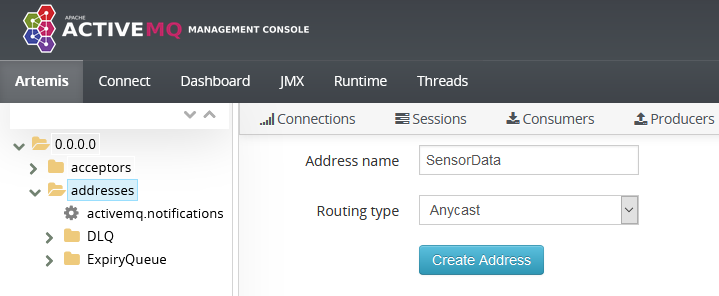
1. Log onto ArtemisMQ management Console: localhost:8161

Username: admin

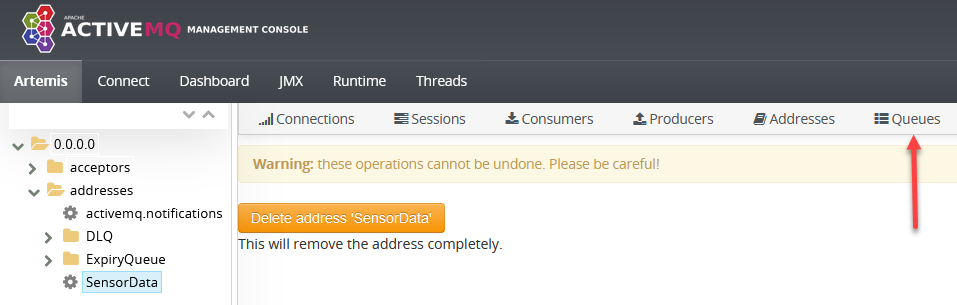
Password: admin

Define Address: SensorData

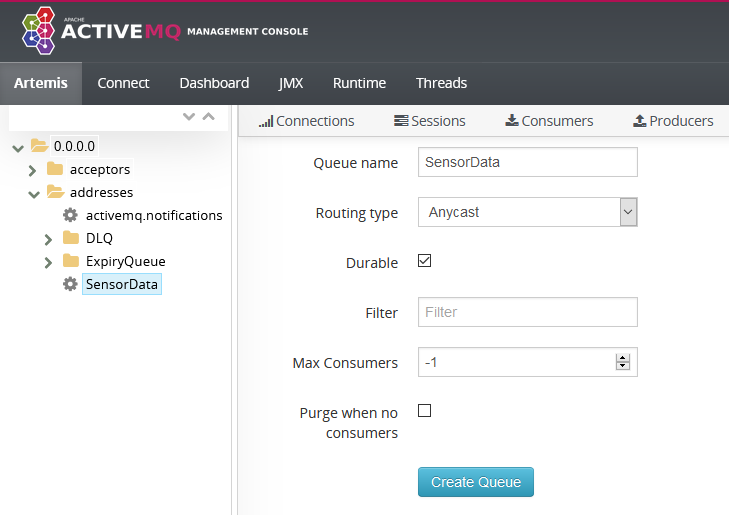
1. Highlight: Addresses folder in the side menu
2. Enter the following:



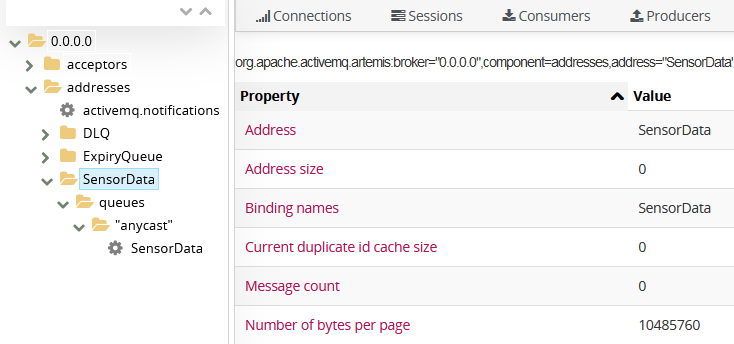
1. Click on the Create Address button.
2. Highlight: SensorData Address and click on: Queue in menu bar.



1. Click on: + Create in top menu bar
2. 
3. Click on: Create Queue button
4. Enter the following:



* You have now defined an endpoint: SensorData bound to an address.
* A Queue: SensorData has been bound to the address.



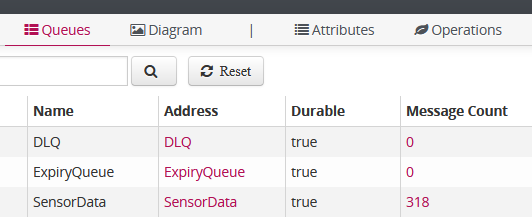
Step 5 – RUN Transformations

Now that everything is place, let’s RUN the Transformations

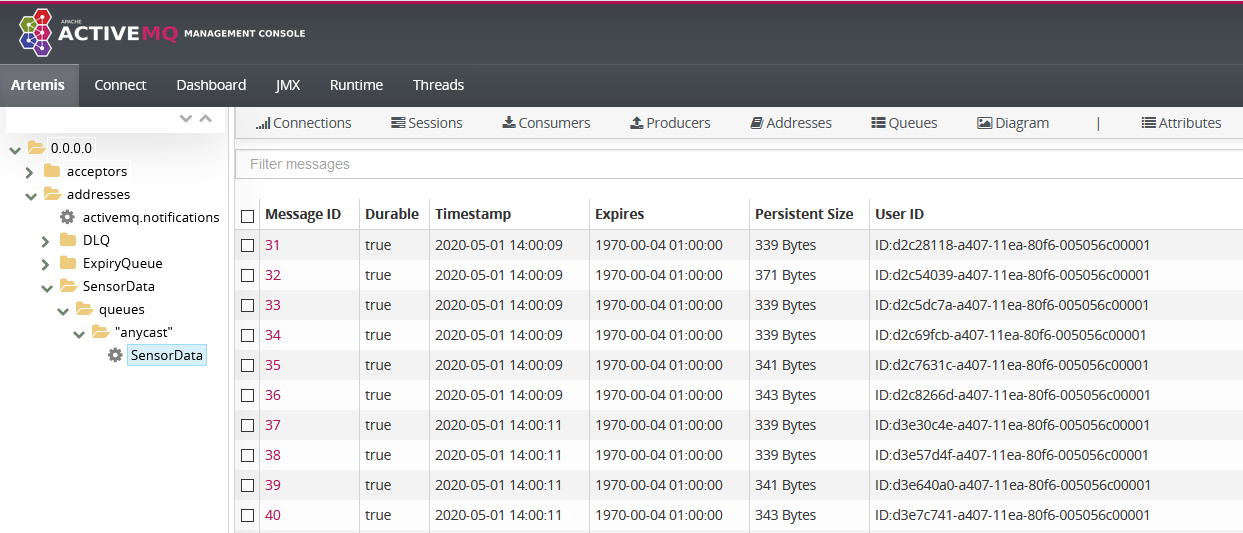
1. Run the transformation: tr\_jms\_producer.ktr

This Publishes truck sensor data to the Queue: SensorData.

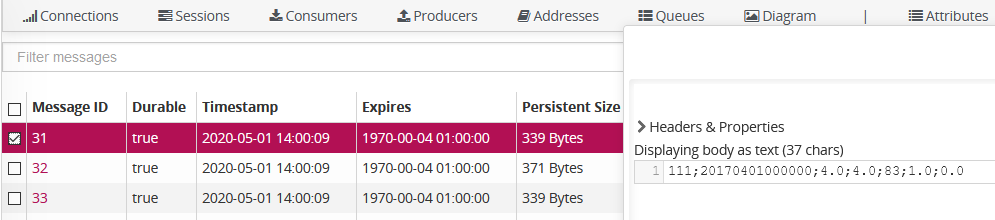
1. Return to the ArtemisMQ management Console
2. Highlight: SensorData Address
3. Click on: Queues option in top menu bar
4. Click on: Message Count for Queue SensorData



* Notice in the overview the Broker has started to receive messages



1. Select one of the messages, then Message ID



Consume the Messages in PDI

1. Run the transformation: tr\_jms\_consumer.ktr

This Consumes truck sensor data from the Queue: SensorData.

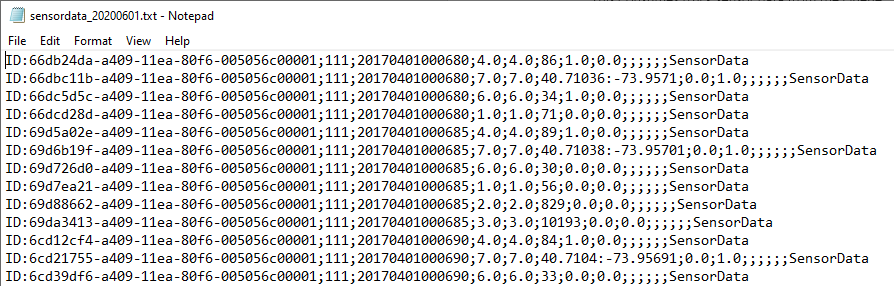
1. Return to the Queue: SensorData

**

*Notice that the messages are immediately consumed. The messages are easily consumed at the same rate they’re being published.*

1. View the output file

C:\ Streaming--Data\03\_JMS\Lab Apache Artemis\output\sensordata\_<datestamp>.txt



*The complete solution can be found at:*

C:\Streaming--Data\03\_JMS\Lab Apache Artemis\solution