Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

# LAB- Understanding streaming Options

In this exercise, you have to work on 02-file-http-streaming-solution project
So, import the corresponding jar given to you

Open db1_product.sql given in src/main/resources and observe it. Execute this .sql file on your mysql8 database (If you don’t have mysql installed, please install it now. Next labs also need mysql)

After executing the .sql file observe that there there are lot of products in products table.

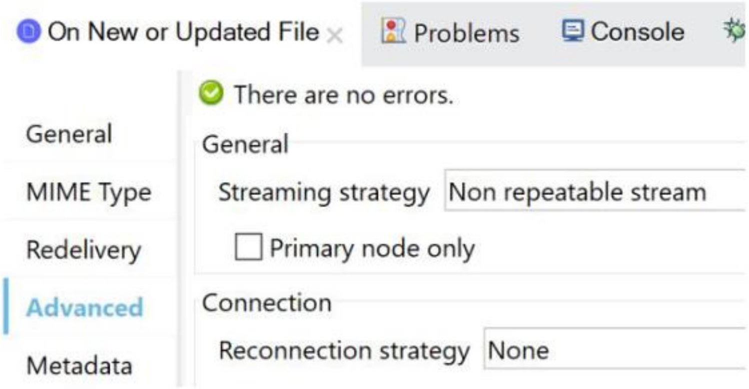
Firstly open file-http-stream.xml and observe 2 flows in it.

First flow is using “On New or update file” which will poll for “c:\files\input” directory for files.

It is calling second flow where scatter-gather is used to write to 2 files concurrently. Scatter gather will try to create 2 copies of Mule message and will invoke 2 routes in this case.

Click on “On New or update file” and select Advanced tab.

Select Stream Strategy as “Non Repeatable Stream” as shown below:



There are no errors.

General

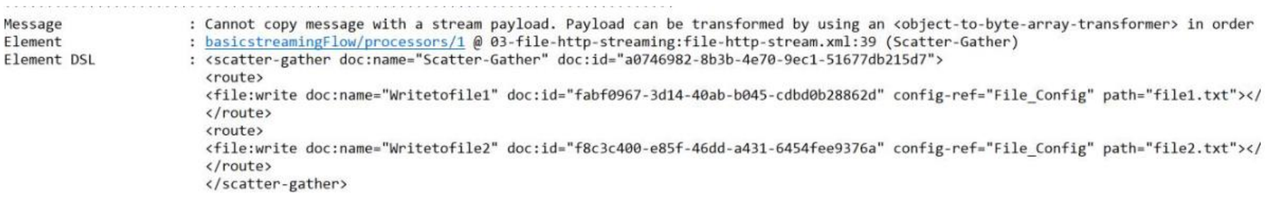
Streaming strategy: Non repeatable stream

□ Primary node only

Connection

Reconnection strategy: None

Now run the application.
Copy sales.csv inside src/main/resources into c:\files\input folder
You should see en error as shown below:



```
Message              : Cannot copy message with a stream payload. Payload can be transformed by using an <object-to-byte-array-transformer> in order
Element              : basicstreamingFlow/processors/1 @ 03-file-http-streaming:file-http-stream.xml:39 (Scatter-Gather)
Element DSL          : <scatter-gather doc:name="Scatter-Gather" doc:id="a0746982-8b3b-4e70-9ec1-51677db215d7">
                       <route>
                       <file:write doc:name="Writetofile1" doc:id="fabf0967-3d14-40ab-b045-cdbd0b28862d" config-ref="File_Config" path="file1.txt"></
                       </route>
                       <route>
                       <file:write doc:name="Writetofile2" doc:id="f8c3c400-e85f-46dd-a431-6454fee9376a" config-ref="File_Config" path="file2.txt"></
                       </route>
                       </scatter-gather>
```

Scatter-Gather cannot create a copy of message if payload is a non repeatable stream. If copy is allowed, 2 concurrent threads (for 2 routes) will read same stream. If it is not a repeatable stream, it will cause problems. That is the reason for this error.

Now change the Streaming strategy as “Repeatable File Store Stream” or “Repeatable in memory stream” and restart the application.

There should be no error and you should see 2 files written with name “file1.txt” and “file2.txt” inside c:\files

If you select “Repeatable in memory stream” what is the importance of “initial buffer size”, “Buffer size Increment”,etc?

If you select “Repeatable File Store Stream”, Why is “in memory size” required?

**Reason : If the file size is less than configured “in memory size” , temp file is also not written to improve performance**

Click on Http Listener and select Advanced tab.

You can select “Streaming Strategy” here. Options are same as the ones we saw for “On new or update file”

Do you understand what is the meaning of “Response Streaming mode” as “AUTO” ?

Open dbstreamdemo.xml and click on “Select” component and click on advanced tab in it.

See the options for “Streaming Strategy”.

Can you tell what is the difference between “Repeatable File Store Iterable” used here and “Repeatable File Store Stream” used in “On new or update file” and Http Listener”.

Explanation: “Select” query return list if records . So, the Stream object returned by this component is internally using a “Iterable” object for the list

“On new Or update file” returns a Stream Object which internally uses “input Stream”.

Don’t get confused. When I say Stream Object is returned by Mule Components, this not the “File Input Stream” or “Http Input Stream”.

Stream Object returned by Mule components internally may use Iterator (Incase of DB Select) or “File Input Stream”(in case of “On new or Update File”) or “Http Input Stream” (Incase of Http Listener)

Now run the application and give a request to http://localhost:8081/test to hit the “dbstreamdemo” flow.

Observe that the logger after “Select” component logs something like below:

```
[[MuleRuntime].uber.03: [03-file-http-streaming].dbstreamdemo.BLOCKING @16c1fe0d]
[processor: dbstreamdemo/processors/1; event:
13466160-9b2b-11ea-82b4-34f39a275da7]
org.mule.runtime.core.internal.processor.LoggerMessageProcessor:
org.mule.runtime.core.internal.streaming.object.ManagedCursorIteratorProvider@6bc322
72
```

So,according the the above log, ManagedCursorIteratorProvider object represents the stream object returned by Select . This object internally uses an iterable object which represents a db curso

# Congratulations!! you understood about streaming options in mule