Batch Processing in Mule 4

[MuleSoft](http://www.mulesoft.com/) allows you to process messages as a batch which can be achieved by batch scope. Batch scope in a mule application can divide the input payload into individual records, performs actions on these individual records, and then sends the processed data to target systems.

Batch job divides payload as 100 records a batch, like that it will process 16 threads at a time.

Batch has three phases in Mule 4

**Load And Dispatch:**

It will create job instances, convert payload into collection of records and then split the collection into individual records for processing.

**Process:**

In thisphase, it processes all individual records asynchronously. Batch step in this phase allows you to filter records.

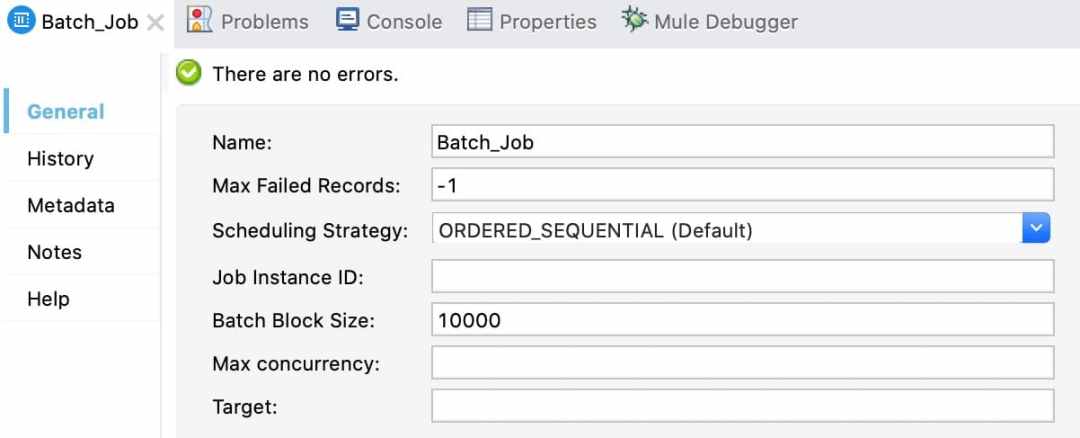
We can also use batch aggregator processor to aggregate records into groups. For example, if you want to process 10 as one group, you can set the aggregate processor size as 10**.**

**On Complete:**

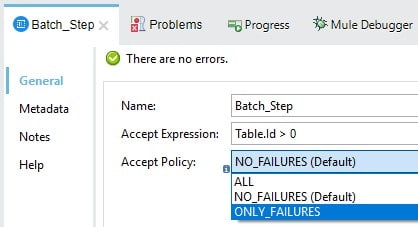
The last and optional phase gives a summary of the payload. It will give us how many records are processed, and how many failed.

One of the options to have millions of records moved from one system to another is by using the **Mule Batch Scope** which makes it possible to handle large data by streaming it from source in smaller chunks of records and processing these asynchronously and reliably. Batch Scope provides several useful features such as-

**Block Sizes** – defines the number of records processed in a step



* **Batch Step** – allows a block of records to be processed sequentially
  + Accept expression configuration – allows you to filter records to be processed in a step by providing conditions
  + **Accept policy configuration** – filtered for error handling



Let’s see an example on Batch Processing.

**PROCESS CSV FILE USING BATCH JOB**

We will process CSV having only 9 records – Sample CSV

Total 9 data records

7 are valid records

2 records will fail – having region as Europe (to show failure scenario)

Batch Job contain following Batch Steps

Batch\_Step\_Process – To process the records as per business needs

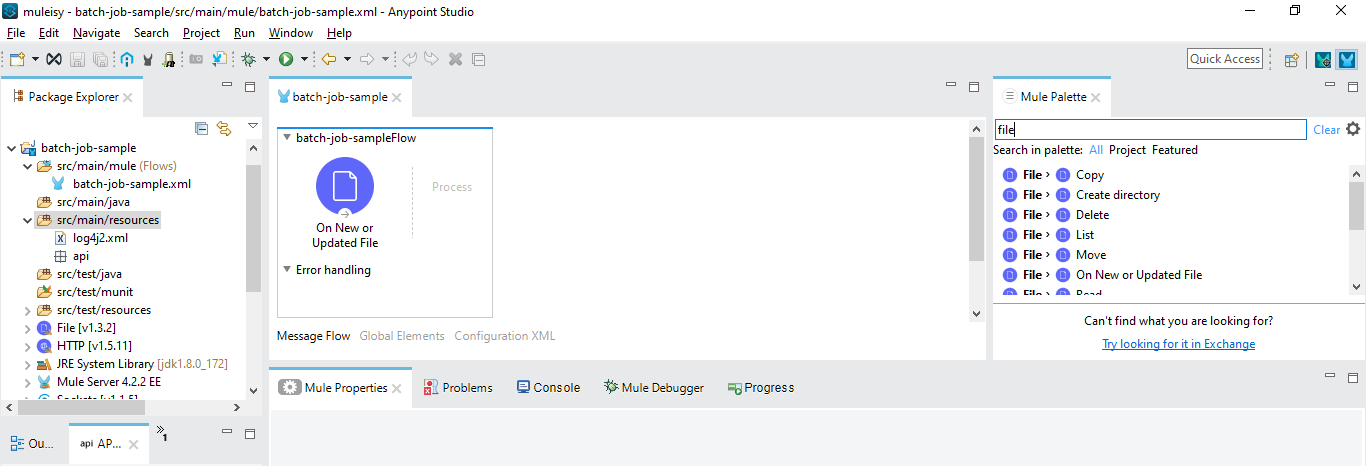
Batch Aggregator\_DB\_Insert – to bulk insert records into DB in batch of 5 records

Batch\_Step\_Failed\_Records – Step to process failed records

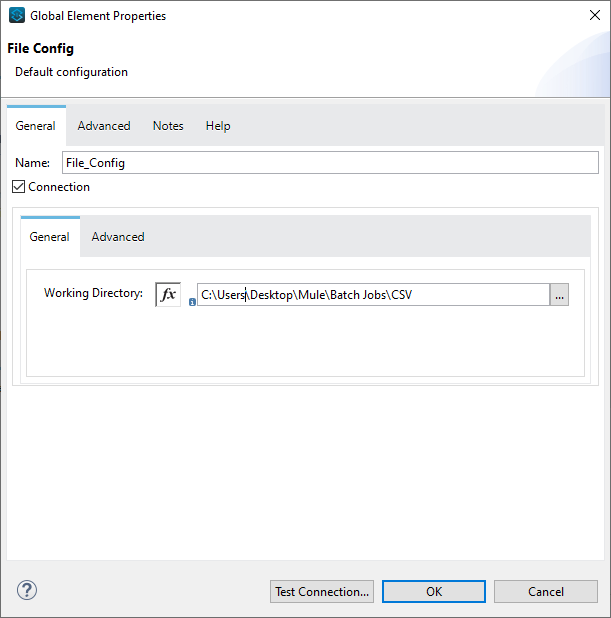
Batch Aggregator Send Email – to send email containing all failed records

**Sample.csv**

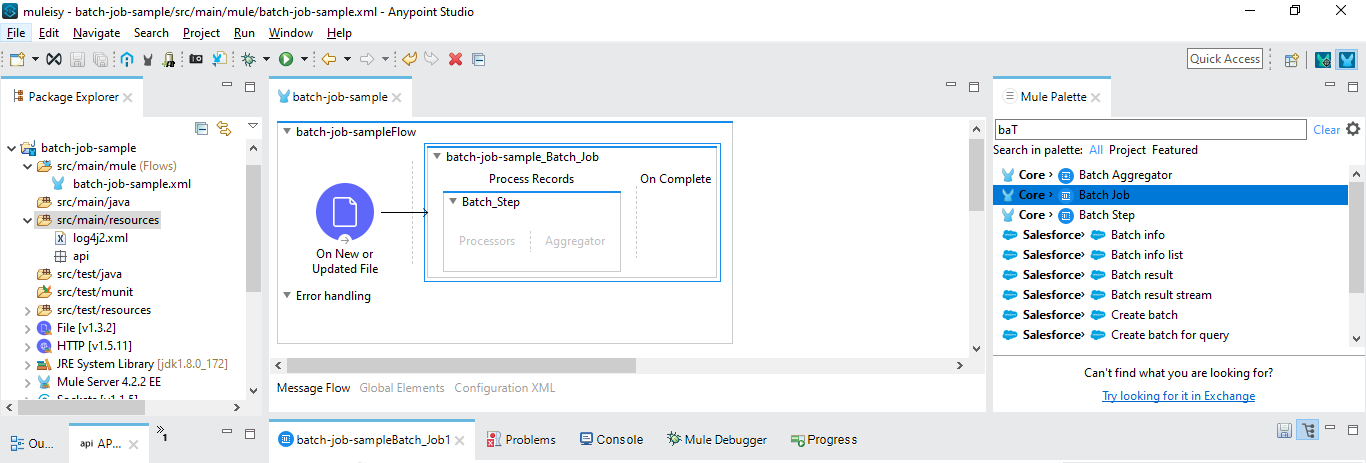
Create a mule application and drag and drop **On New or Updated File** from File connector – this will monitor the folder for any new or updated file



Add the File Config

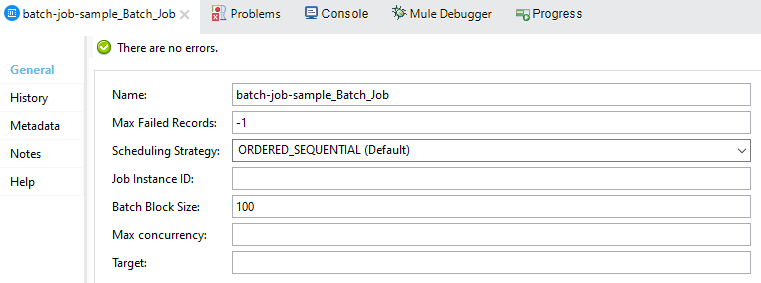


Drag and drop Batch Job from Core activities



Batch Job will have following configuration

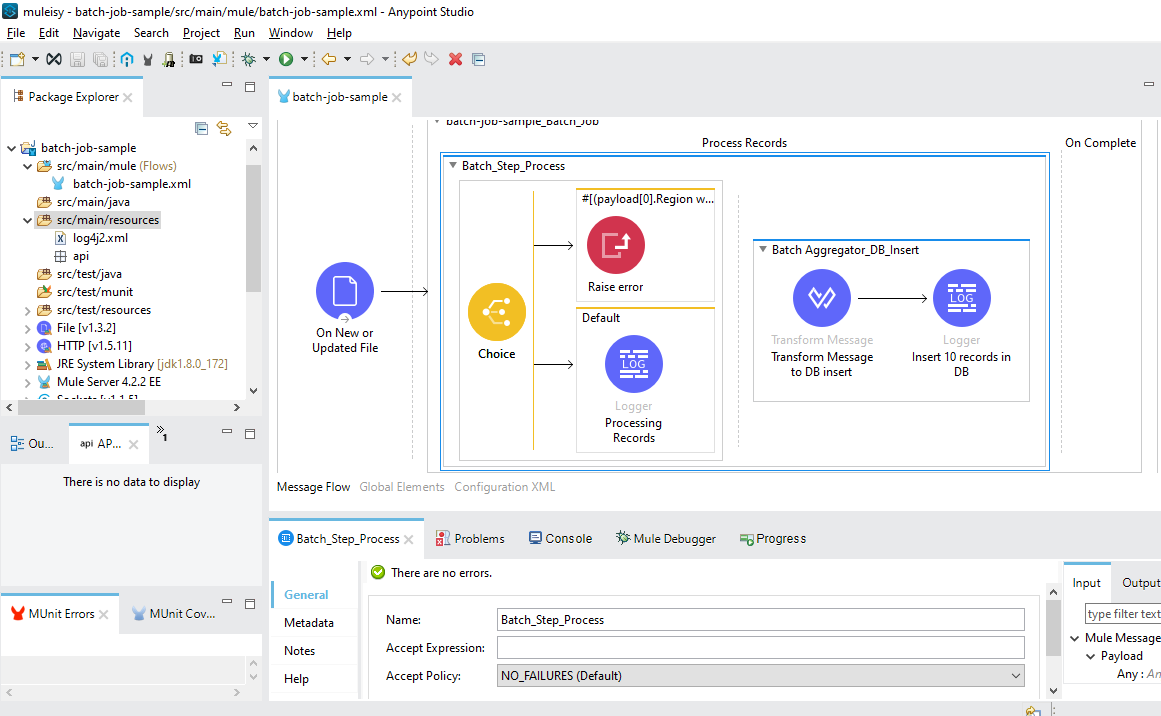
* Will set the **Max Failed records to -1** as we want to process whole CSV file



Now we will add the batch step to do following processing

* Accept **All or No failures** records as accept policy
* If Region in CSV has a value of Europe then we will raise a custom exception – added choice
* Otherwise log the message – **processing #[payload]**
* Add Batch Aggregator
  + Set the **aggregator size to 5**
  + Transform the payload collection of 5 records to DB insert request
  + Log the message – **insert into DB #[payload]**

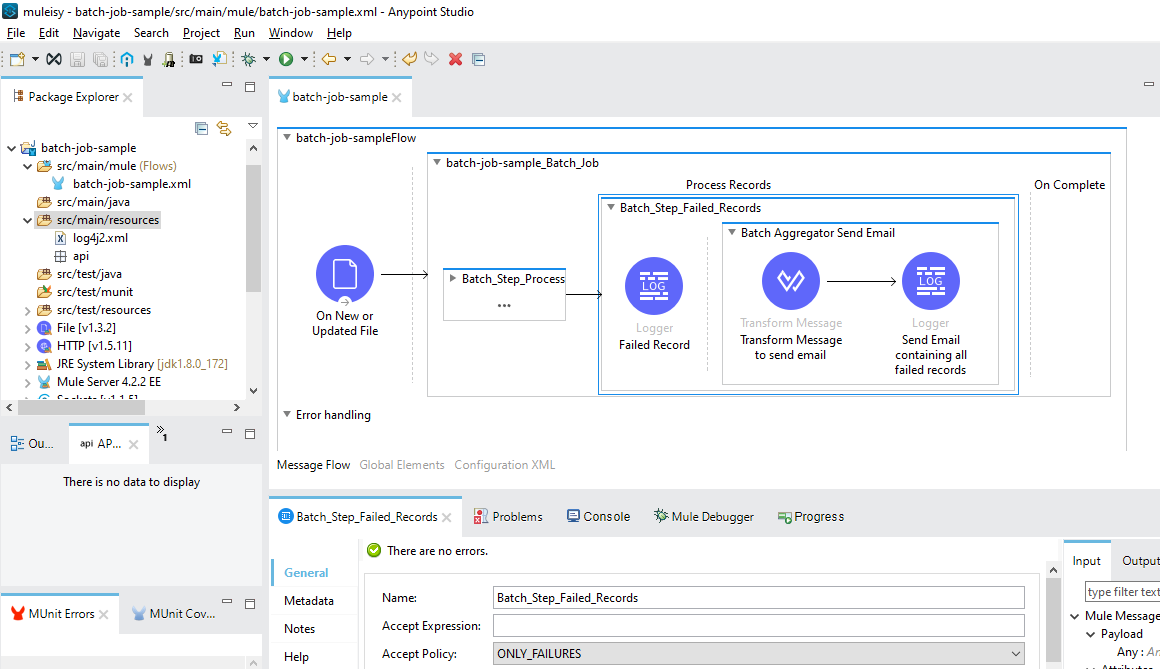
So the updated batch step will look below



Now we will add another batch step to handle all the failure records

* Accept **Only Failures records** as accept policy
* Logs the message –**failed record found #[payload]**
* Add Batch Aggregator
  + Set the **aggregator size to Streaming** (means for all the failed records)
  + Transform the payload to email payload
  + Log the message –**sending email for records #[payload]**

So the batch step for processing failed records will look like below



And in last will add log in On Complete phase

* Log message – **summary report – #[payload write “application/json”]**

**Key Pointers for using Batch efficiently in MuleSoft:**

1. Transformation complexity – Use transform before batch step and avoid dataweave in process batch steps as it will process one record at a time which is inefficient and doesn’t justifiy the use of batch processing
2. No. of Batch Steps – Dividing the process into steps makes it easier to isolate a failed batch and have it reprocessed separately
3. Block Sizes – Running comparative tests with different values and testing performance helps you find an optimum block size before moving this change into production. Modifying this value is optional. If no changes are applied, the default value is 100 records per block.
4. Scheduling Strategy – It enables you to control how instances of a given batch job are executed. The default configuration is ORDERED\_SQUENTIAL which is suitable If several job instances are in an executable state at the same time, the instances execute one at a time based on their creation timestamp. The other setting available is ROUND\_ROBIN which attempts to execute all available instances of a batch job using a round-robin algorithm to assign the available resources

 More detail on Batch Job – <https://docs.mulesoft.com/mule-runtime/4.3/batch-processing-concept>