



TGH

Making Integrations Simpler

boomi
Partner



Sorting of documents based on a particular field

Author
Apurba Sarkar



Sorting of documents based on a particular field

In this blog we will perform sorting of a document based on a particular field, sample input and output documents are given below:

Input:

```
{
  "Employees" :[
    {
      "Name" : "Jack",
      "Sequence" : 3
    },
    {
      "Name" : "Jack",
      "Sequence" : 5
    },
    {
      "Name" : "Tony",
      "Sequence" : 2
    },
    {
      "Name" : "Peter",
      "Sequence" : 4
    },
    {
      "Name" : "Steve",
      "Sequence" : 10
    }
  ]
}
```

Expected Output:

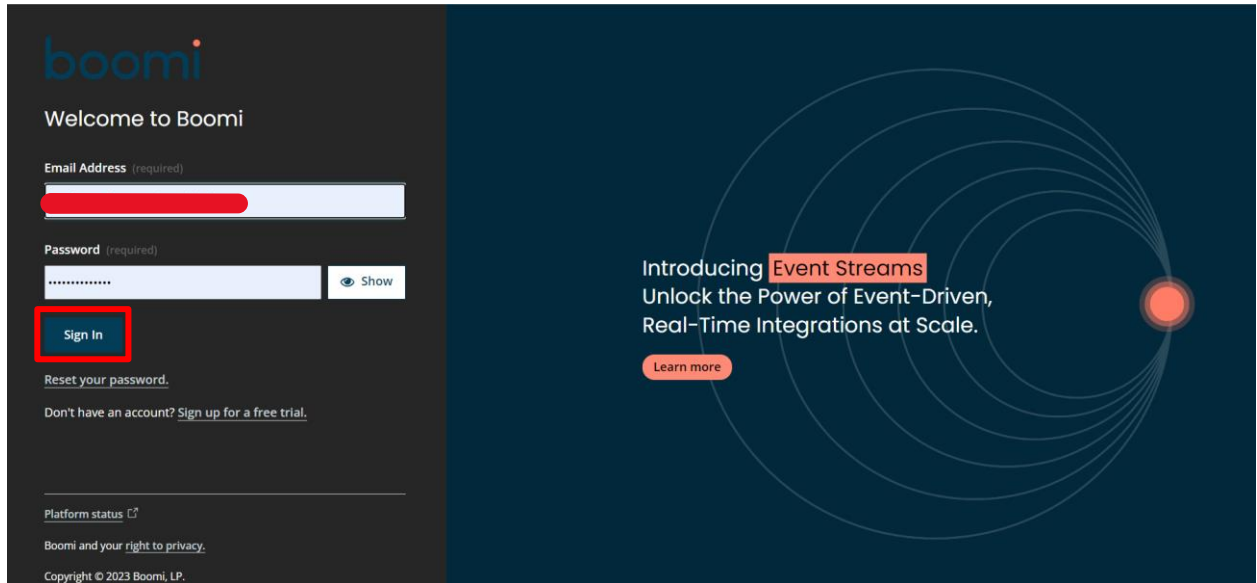
```
{
  "Employees" :[
    {
      "Name" : "Tony",
      "Sequence" : 2
    },
    {
      "Name" : "Jack",
      "Sequence" : 3
    },
    {
      "Name" : "Peter",
      "Sequence" : 4
    },
    {
      "Name" : "Steve",
      "Sequence" : 10
    }
  ]
}
```

Overview of the usecase:

1. We will read the document mentioned in the input section using the Disk V2 connector.
2. We will convert the document into a Flat file and use the Enforce Unique option in the Flat file profile to remove the duplicates(In case we have duplicates in our document).
3. We will use one Groovy script to perform the sorting.
4. Finally we will convert the document back to JSON format.

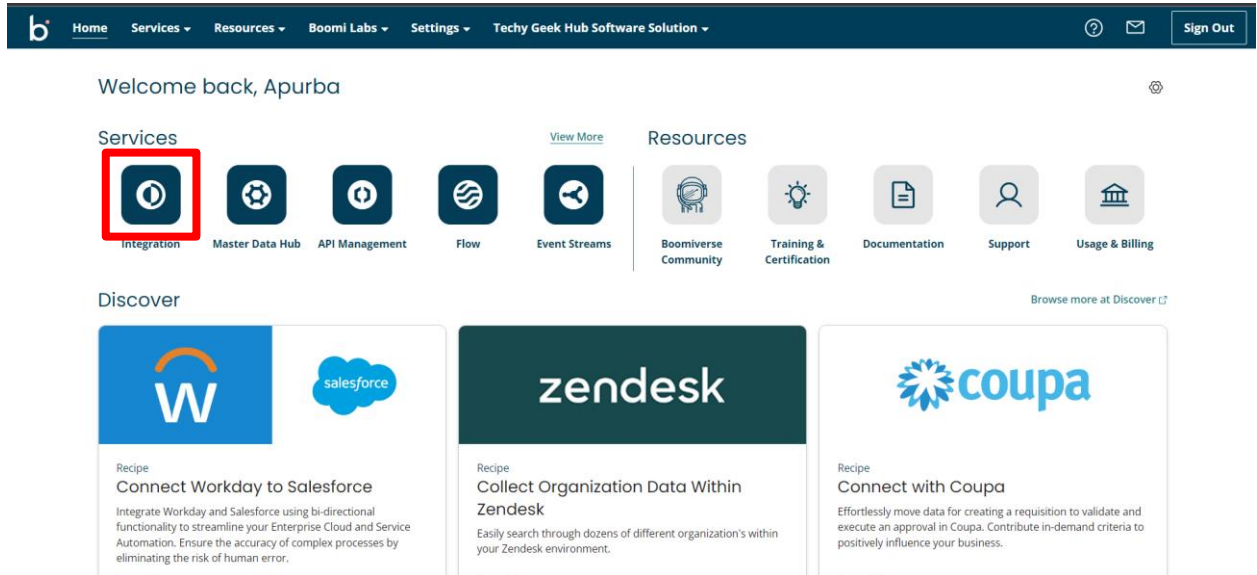
Steps:

Step 1. Go to platform.boomi.com



Step 2. Login by giving your valid credentials (Username and Password).

Step 3. Home page will be displayed, now click on “Integration”.

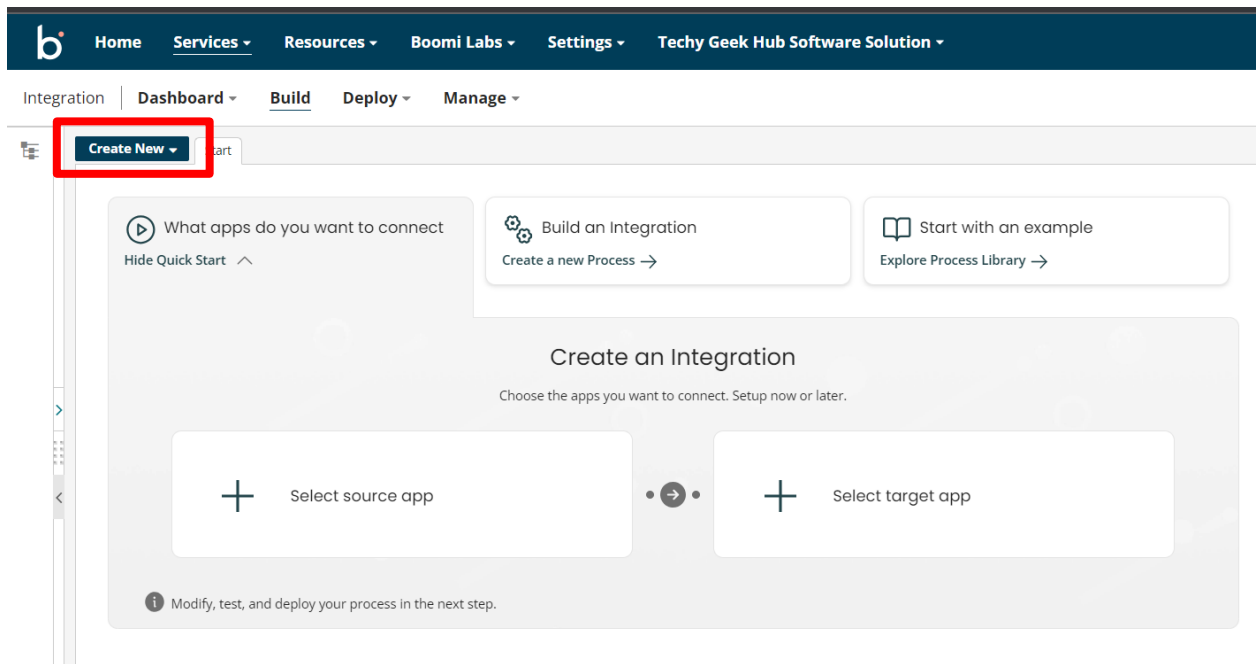


©[TGH Software Solutions Pvt. Ltd.](https://www.techygeekhub.com)

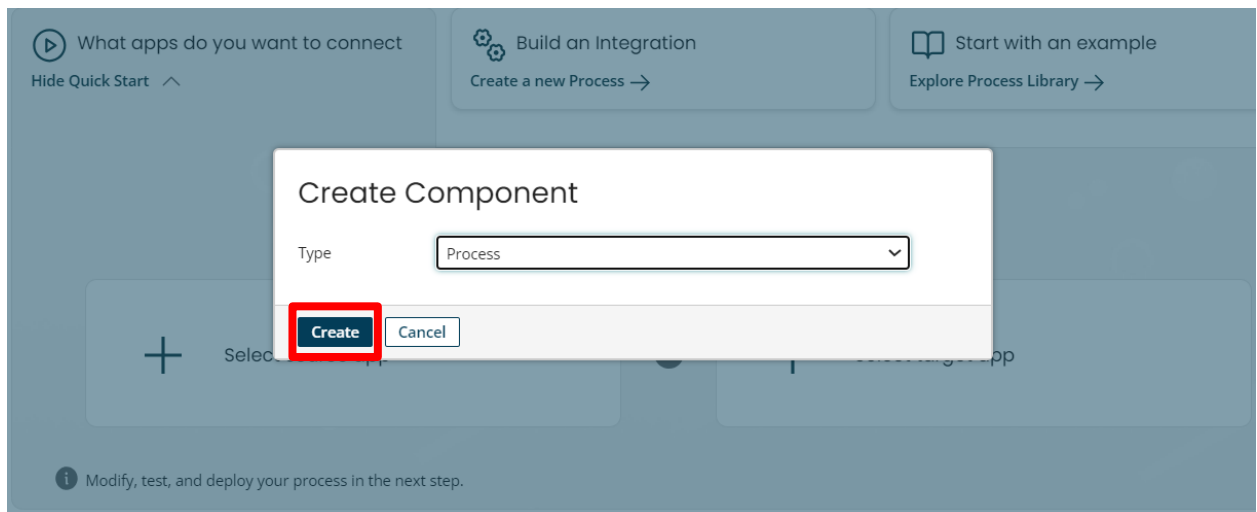
No part of this document may be copied, reproduced, republished, uploaded, posted, publicly displayed, encoded, translated, transmitted or distributed in any way to any other computer, server, website or other medium for publication or distribution, without TGH's prior written consent



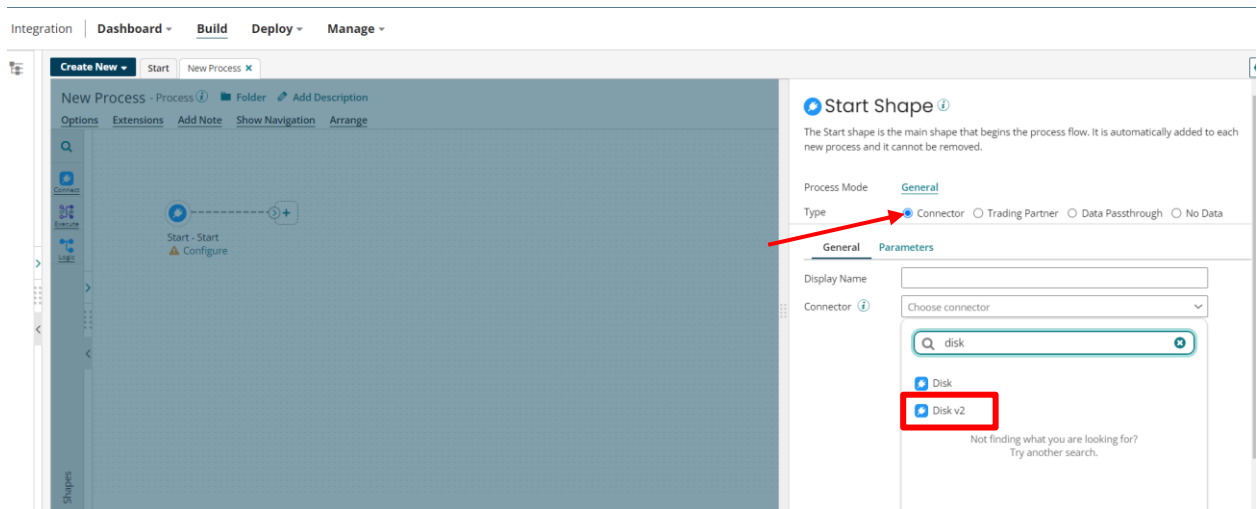
Step 4. Click on the “Create New” button.



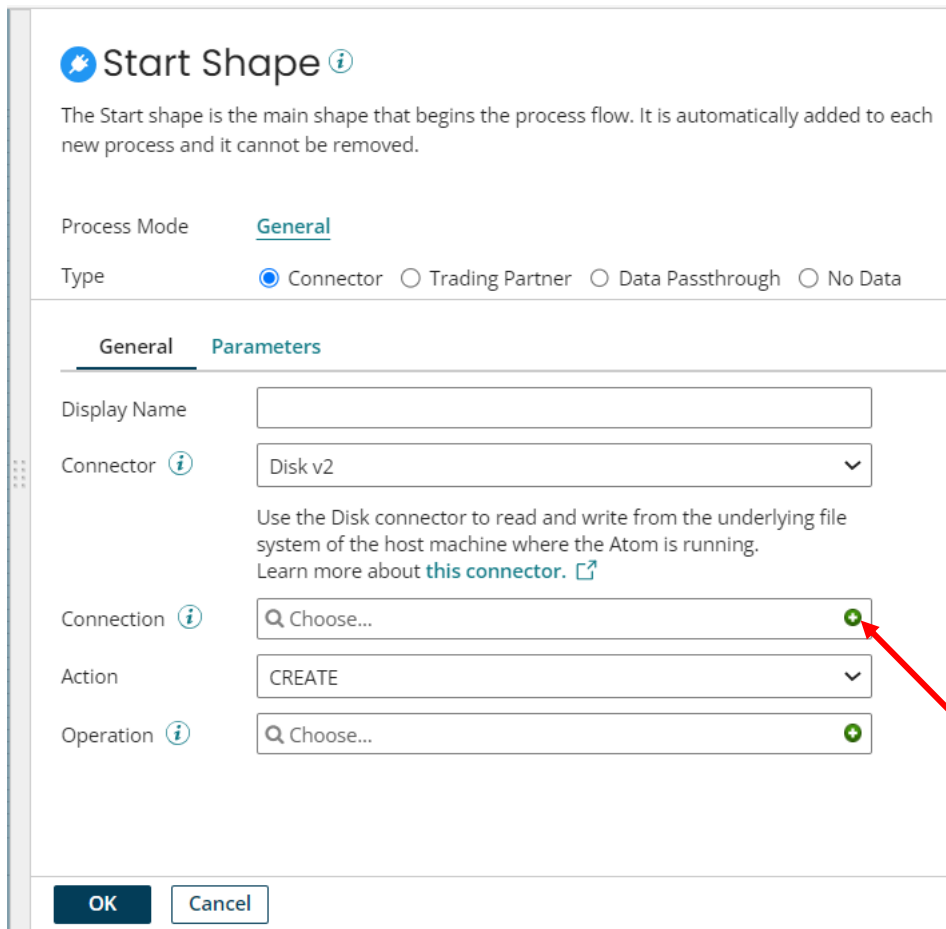
Step 5. Select “Process” and click on “Create”.



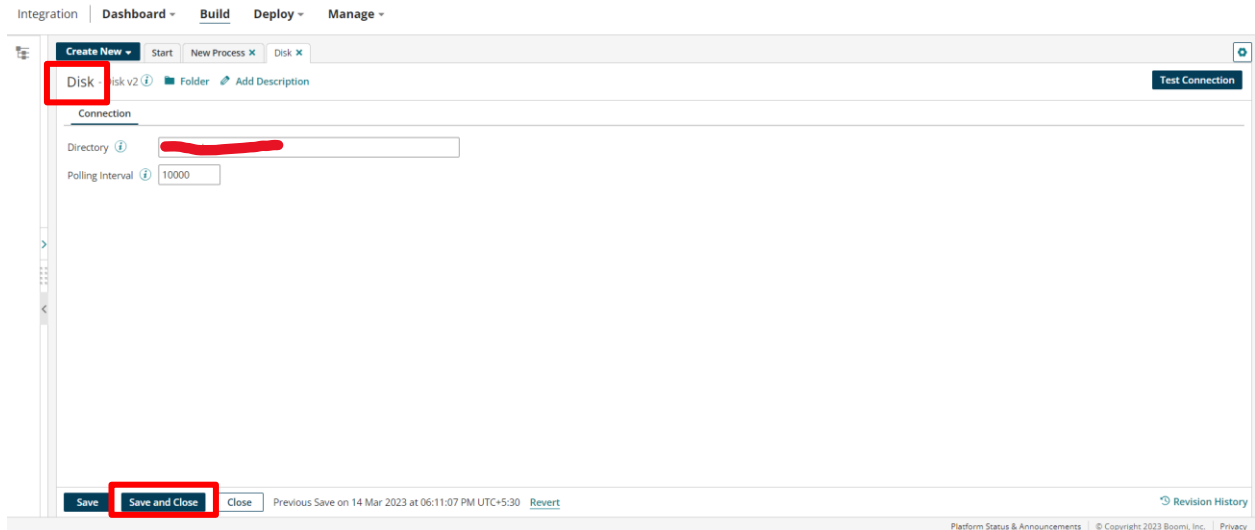
Step 6. Choose Start shape as “Connector” type and choose connector as “Disk V2”



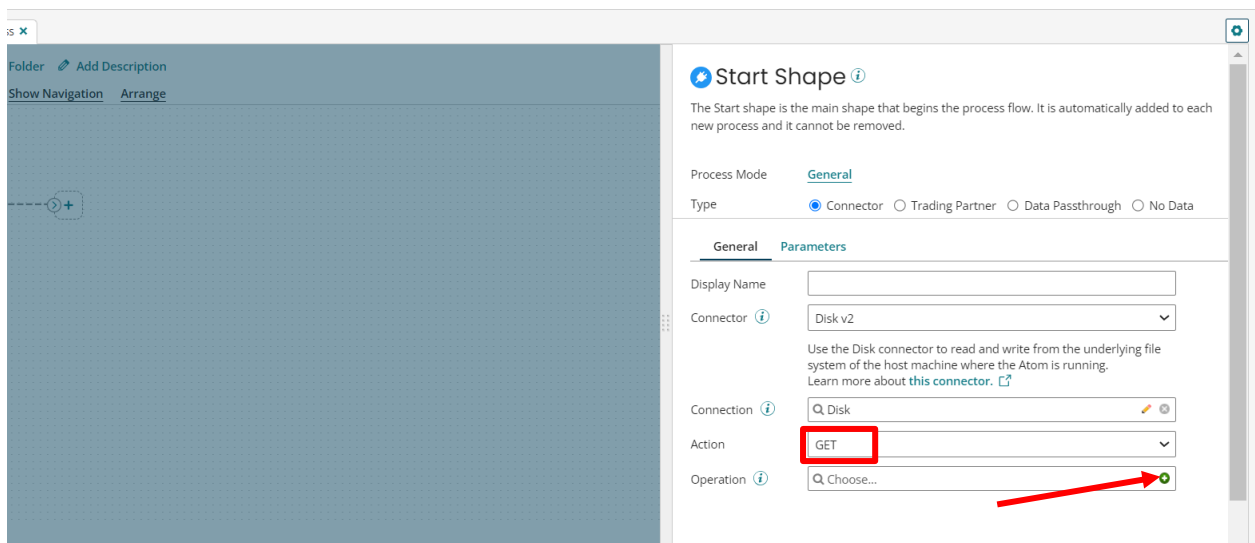
Step 7. Click on ‘+’ icon in the Connection to configure the connection.



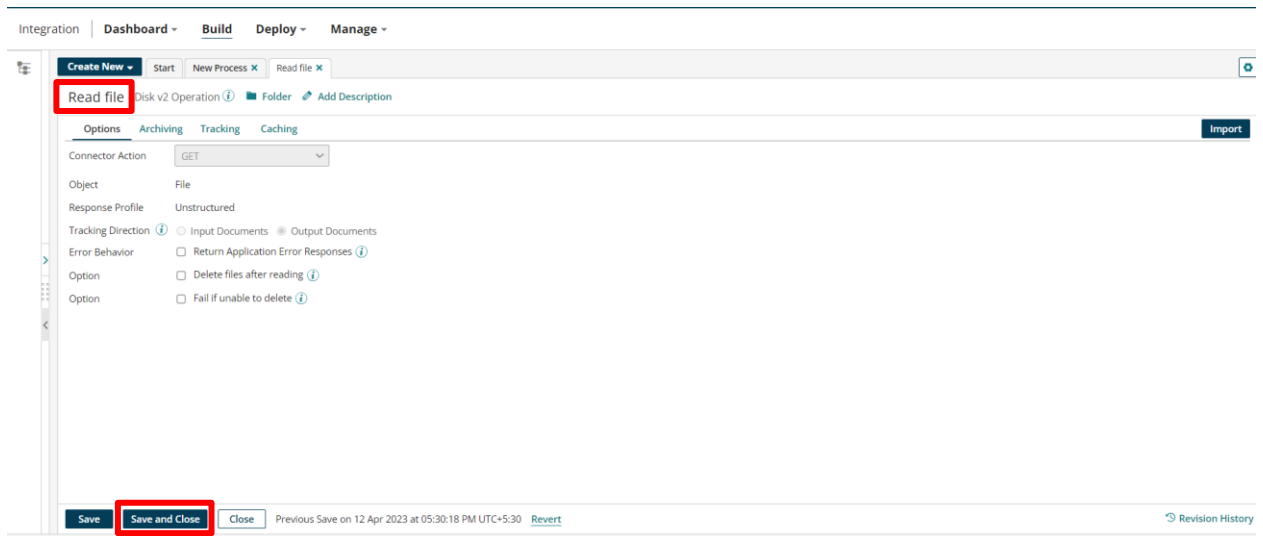
Step 8. Give the connection name as 'Disk', in the directory text field give the directory path from where you want to read the file, at last click 'Save and Close'.



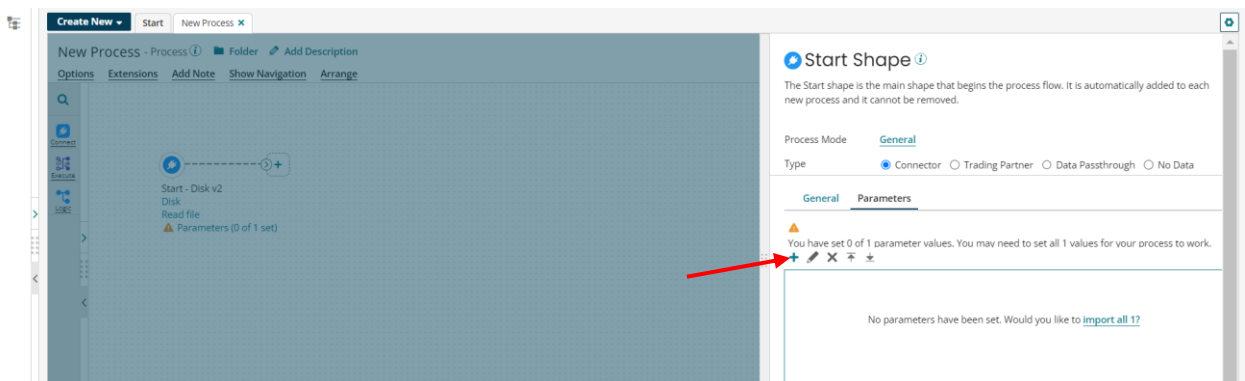
Step 9. Select action as "Get" and Click on "+" icon in Operation to configure operation.



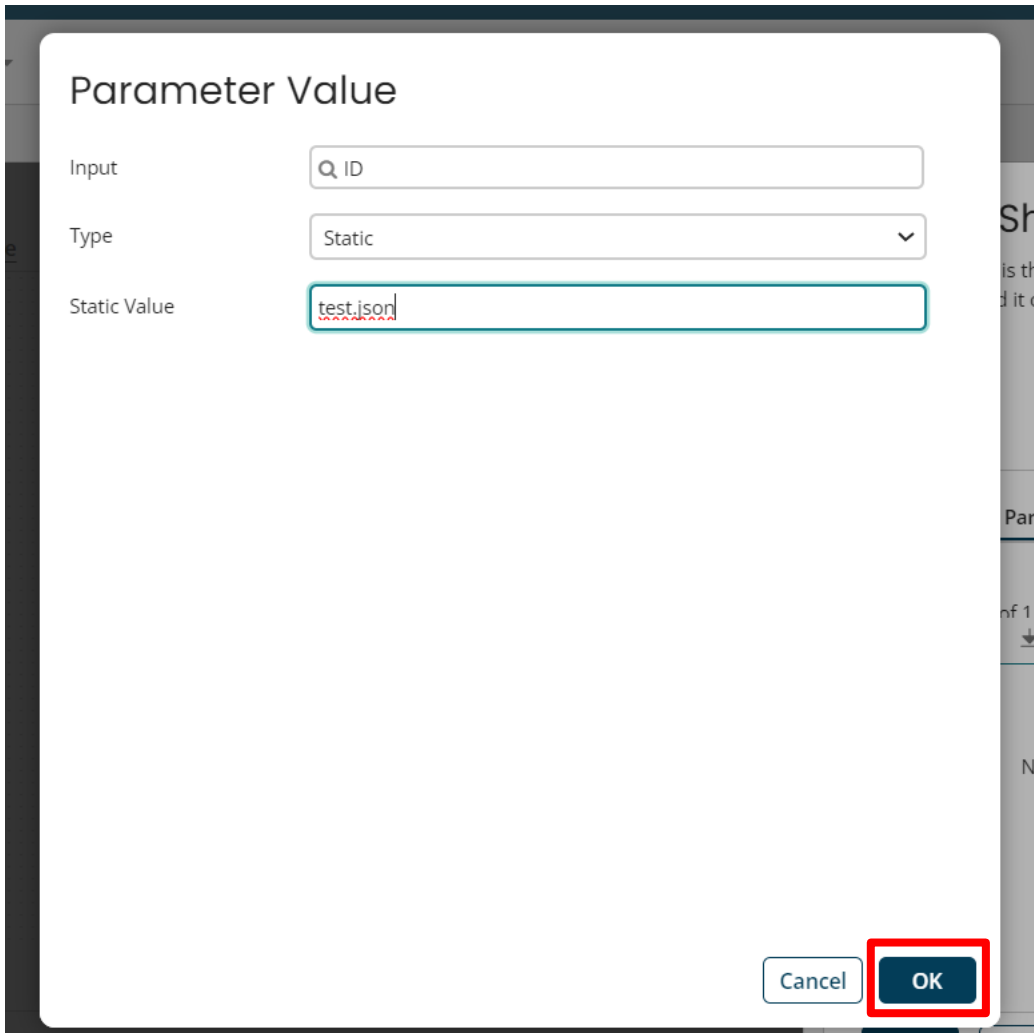
Step 10. Give the operation name as “Read file” and click on “Save and Close”.



Step 11. Click on Parameter, then click on the “+” button.



Step 12. Select Input as ID, give the file name(test.json) in static value and hit on “Ok”.



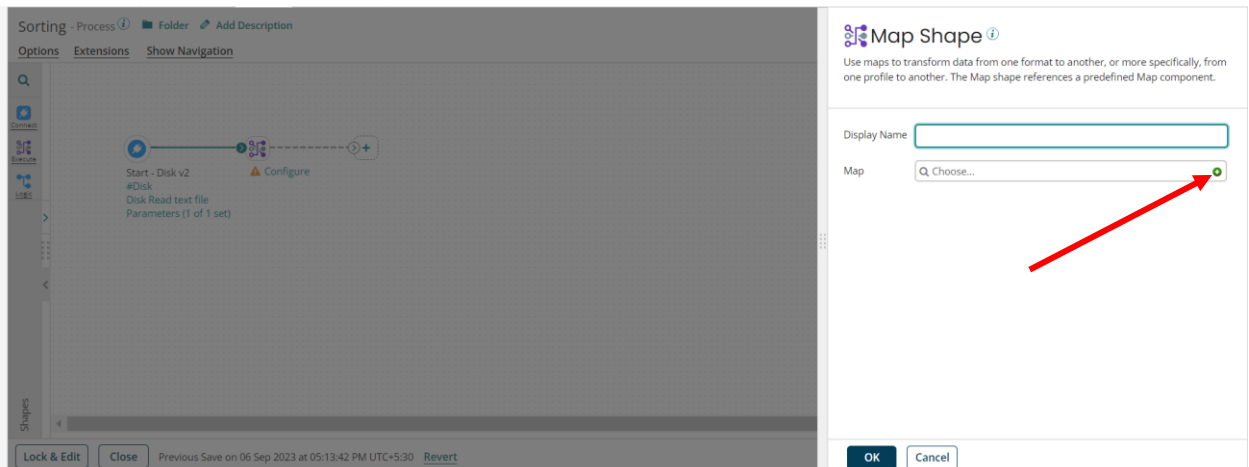
Parameter Value

Input

Type

Static Value

Step 13. Drag and drop a Map shape and click on “+” icon.



Sorting - Process

Options Extensions Show Navigation

Start - Disk v2

#Disk
Disk Read text file
Parameters (1 of 1 set)

Map Shape

Use maps to transform data from one format to another, or more specifically, from one profile to another. The Map shape references a predefined Map component.

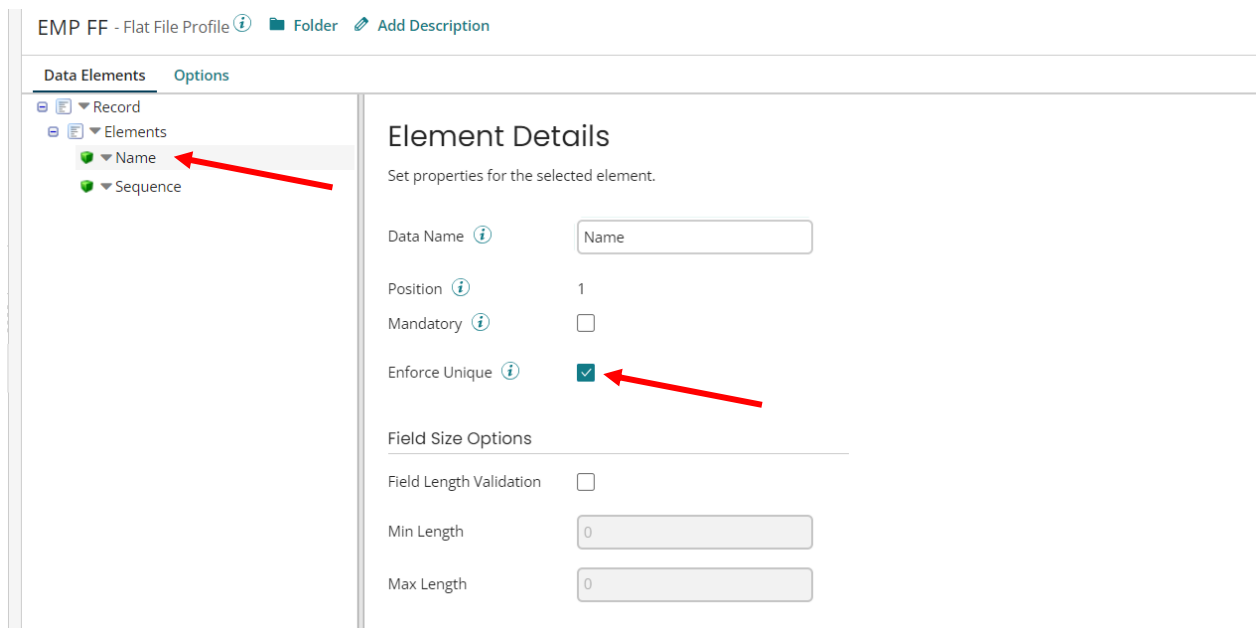
Display Name

Map

Step 14. Choose the JSON profile in the Source.



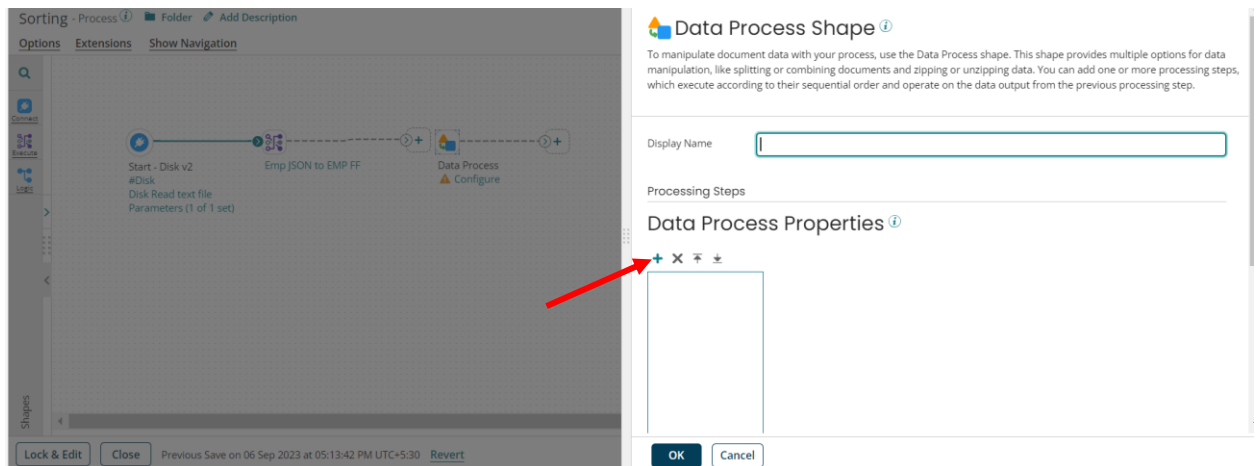
Step 15. Create one Flat file also for our input. Make sure the Enforce Unique check box should be checked in Name element.



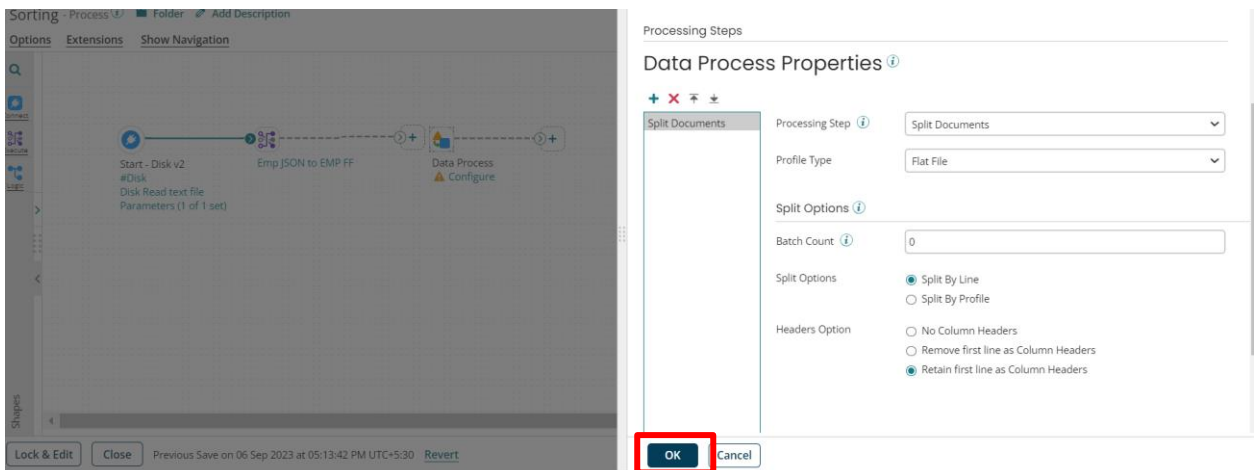
Step 16. Map all the elements and click “Save and close”.



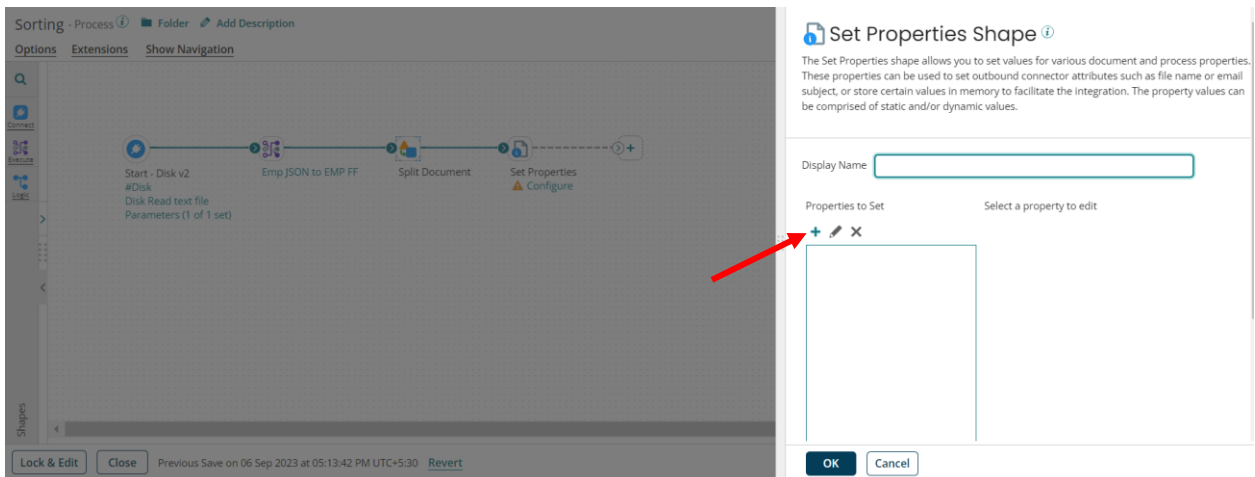
Step 17. Drag and drop a Data process shape and click on the “+” icon to add a processing step.



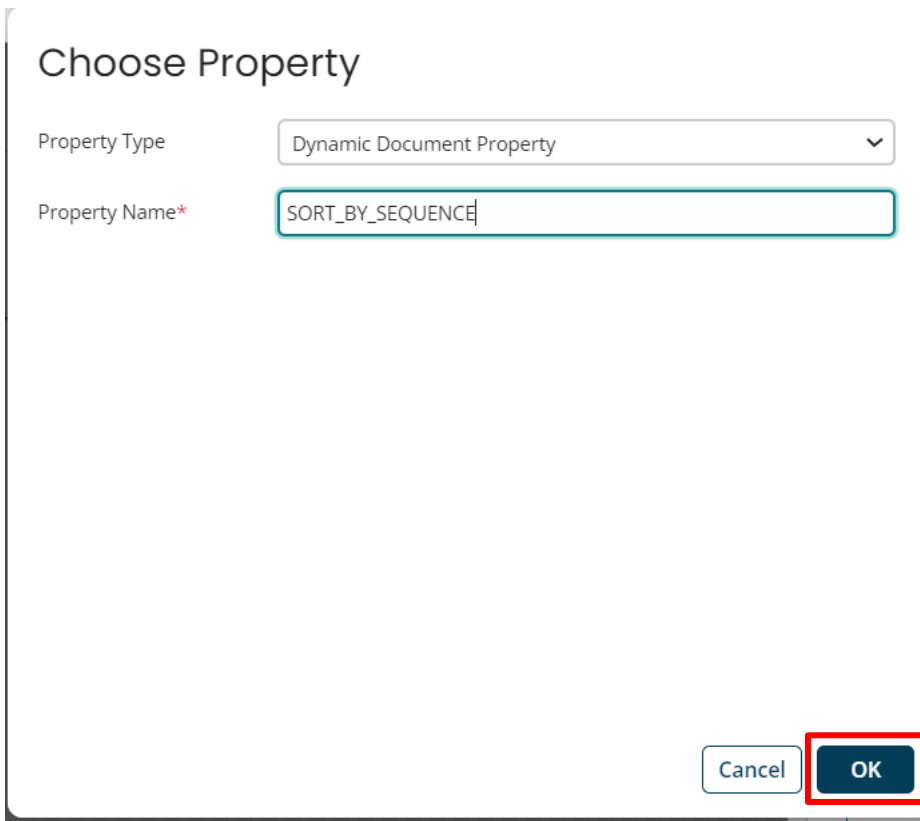
Step 18. Select Spil Documents, choose Profile Type as Flat file, Split option as Split By Line as shown below:



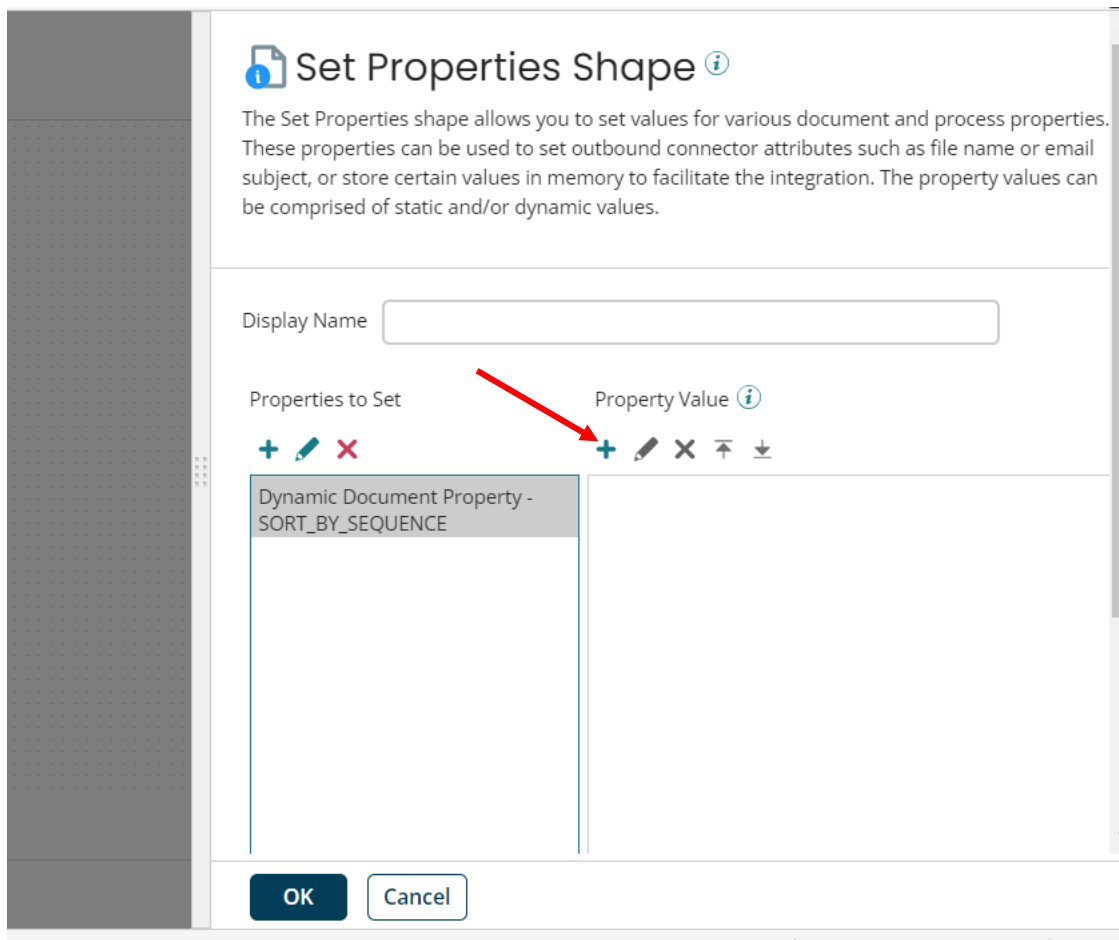
Step 19. Drag and drop a Set Properties shape and click on the “+” icon to add a property.



Step 20. Select property type as Dynamic Document Property and give the property name “SORT_BY_SEQUENCE” and click on “OK”.



Step 21. Click on “+” icon to add Property value.



Step 22. Choose Type as Profile Element, choose the EMP FF profile and in Element choose the Sequence element. Click on “Ok”.

Parameter Value

Type: Profile Element

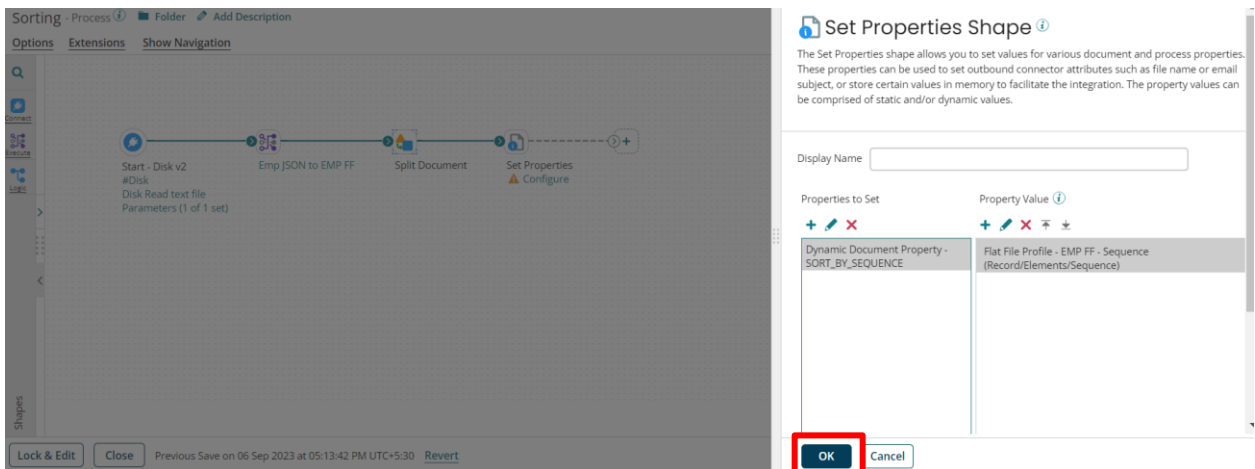
Profile Type: Flat File

Profile: EMP FF

Element: Sequence (Record/Elements/Sequence)

Cancel OK

Step 23. It will look like this, now click on “Ok”



Set Properties Shape

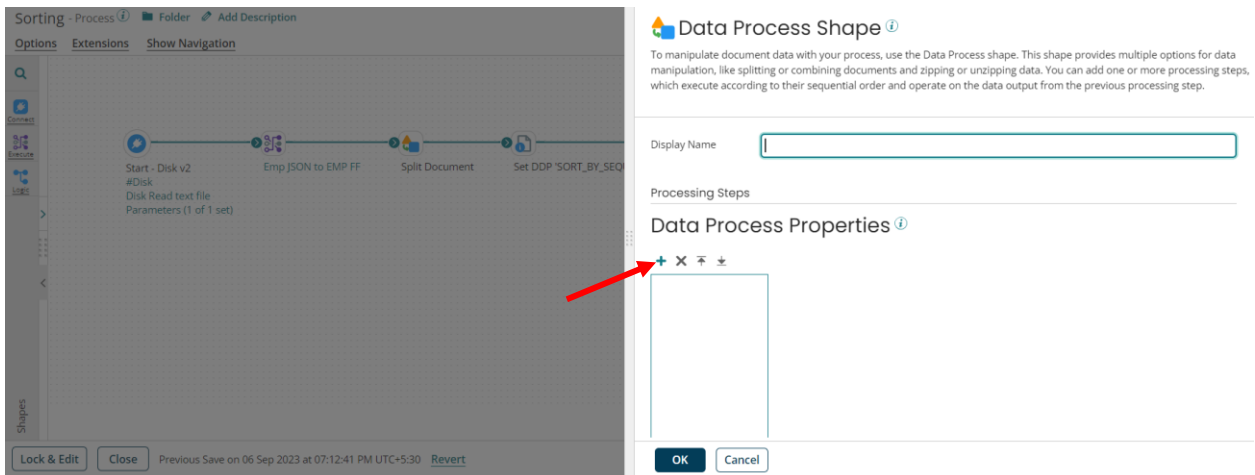
The Set Properties shape allows you to set values for various document and process properties. These properties can be used to set outbound connector attributes such as file name or email subject, or store certain values in memory to facilitate the integration. The property values can be comprised of static and/or dynamic values.

Display Name:

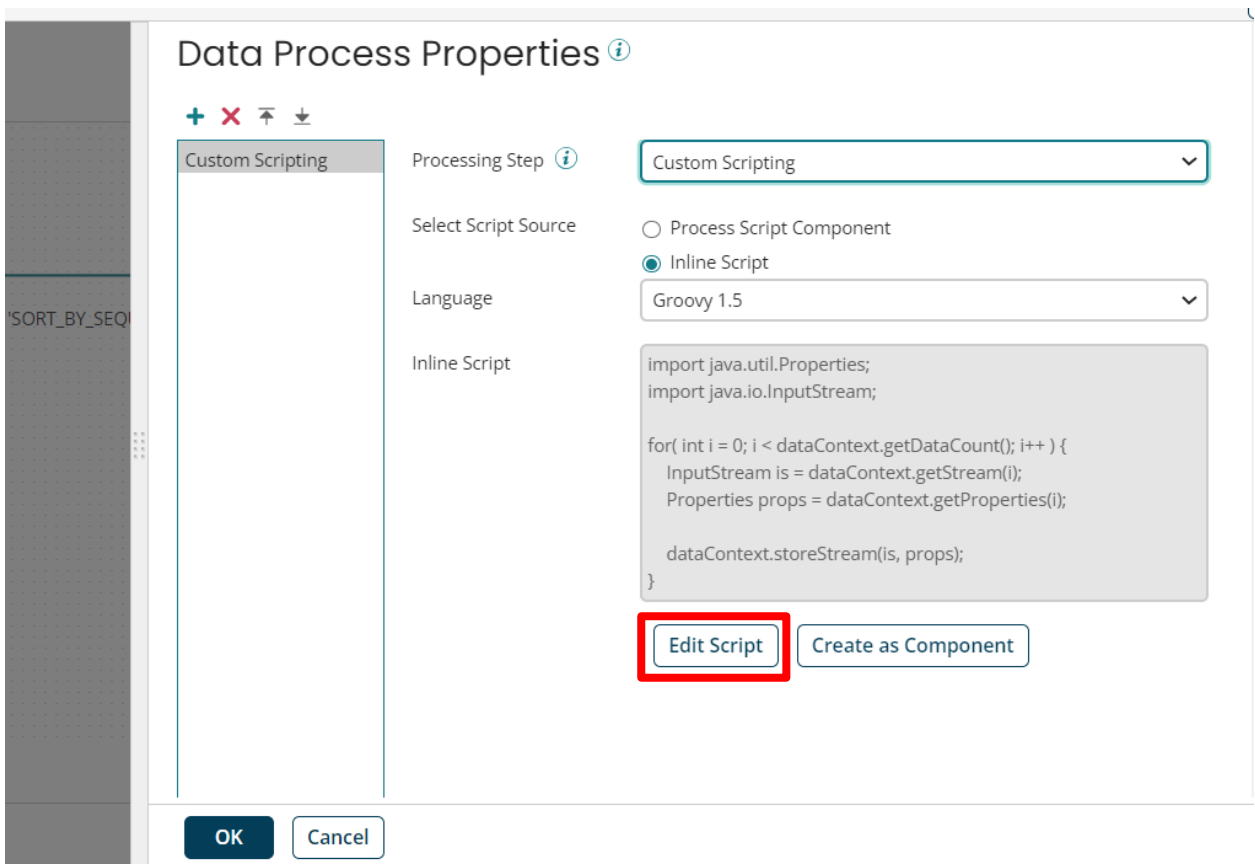
Properties to Set	Property Value
Dynamic Document Property - SORT_BY_SEQUENCE	Flat File Profile - EMP FF - Sequence (Record/Elements/Sequence)

OK Cancel

Step 24. Drag and drop one Data Process shape and click on the “+” icon to add a processing step.



Step 25. Choose Custom Scripting as Processing Step, Script Source is Inline Script and Language is Groovy 1.5. Now click on “Edit Script” to write the script.



Step 26. Write this scripting for performing sorting.

```

1 import java.util.Properties
2 import java.io.InputStream
3
4 int documentCount = dataContext.getDataCount()
5 def valueArray = []
6 def indexArray = []
7
8 // Loop through documents and store the sort-by-sequence and document indices in the valueArray and indexArray.
9 for (int i = 0; i < documentCount; i++) {
10   Properties props = dataContext.getProperties(i)
11   int sortByValue = props.getProperty("document.dynamic.userdefined.SORT_BY_SEQUENCE").toInteger();
12   valueArray.add(sortByValue)
13   indexArray.add(i)
14 }
15
16 def temp
17
18 // Bubblesort
19 for (int i = 0; i < documentCount; i++) {
20   for (int j = 0; j < documentCount - 1; j++) {
21     if (valueArray[j] > valueArray[j + 1]) {
22
23       temp = valueArray[j]
24       valueArray[j] = valueArray[j + 1]
25       valueArray[j + 1] = temp
26
27       temp = indexArray[j]
28       indexArray[j] = indexArray[j + 1]
29       indexArray[j + 1] = temp
30     }
31   }
32 }
33
34 // Output sorted documents
35 for (int l = 0; l < documentCount; l++) {
36   int m = (int) indexArray[l]
37   dataContext.storeStream(dataContext.getStream(m), dataContext.getProperties(m))
38 }

```

Groovy Guide

All data is referenced from the **dataContext** object, which contains the following methods:

- getDataCount()** - Gets the number of documents that are being processed.
- getStream(int)** - Gets an InputStream for a given document index.
- getProperties(int)** - Gets a Properties object for a given document index.
- storeStream(InputStream, Properties)** - Stores the data located in the InputStream back to the process, along with any Properties.

Data should be grabbed by the **getStream(int)** method, manipulated, then stored back to the **dataContext** using **storeStream(InputStream, Properties)**

Note – Here is a small demonstration of the above scripting:

//import the necessary libraries.

```
import java.util.Properties
```

```
import java.io.InputStream
```

//Store the document count in documentCount.

```
int documentCount = dataContext.getDataCount()
```

//Create two arrays, one for storing the Sequence values and other one for storing the indexes for them.

```
def valueArray = []
```

```
def indexArray = []
```

// Loop through documents and store the sort-by-sequence and document indices in the valueArray and indexArray.

```
for (int i = 0; i < documentCount; i++) {
```

```
    Properties props = dataContext.getProperties(i)
```

```
    int sortByValue =
```

```
props.getProperty("document.dynamic.userdefined.SORT_BY_SEQUENCE").toInteger();
```

```
    valueArray.add(sortByValue)
```

```
    indexArray.add(i)
```

```
}
```

©TGH Software Solutions Pvt. Ltd.


```
// Performing Bubble sort using two for loops and using 3rd variable (temp).
```

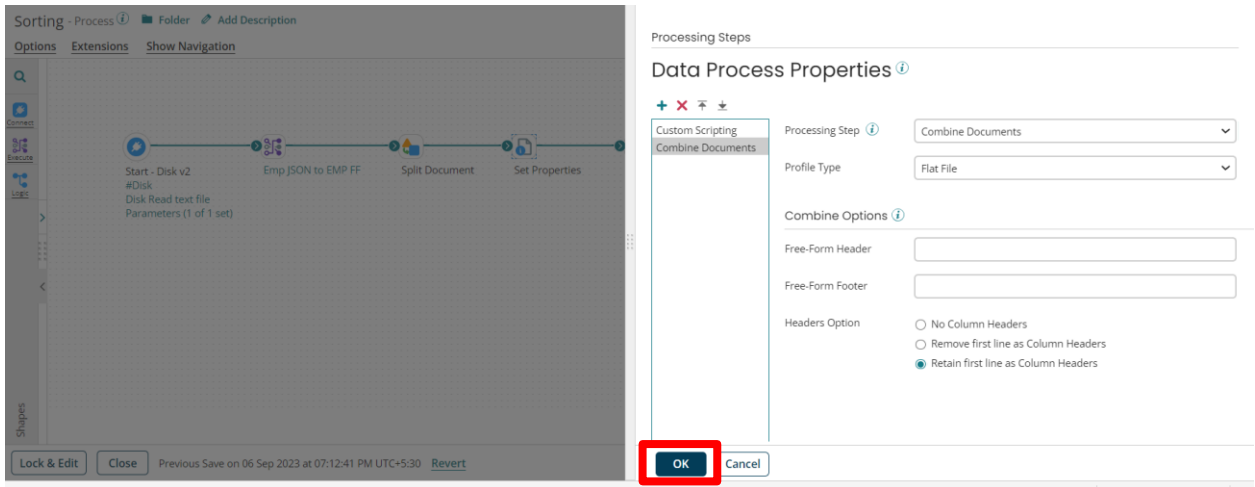
```
def temp
```

```
for (int i = 0; i < documentCount; i++) {  
  for (int j = 0; j < documentCount - 1; j++) {  
    if (valueArray[j] > valueArray[j + 1]) {  
  
      temp = valueArray[j]  
      valueArray[j] = valueArray[j + 1]  
      valueArray[j + 1] = temp  
  
      temp = indexArray[j]  
      indexArray[j] = indexArray[j + 1]  
      indexArray[j + 1] = temp  
    }  
  }  
}  
  
// Output sorted documents back in the process.  
for (int l = 0; l < documentCount; l++) {  
  int m = (int) indexArray[l]  
  dataContext.storeStream(dataContext.getStream(m), dataContext.getProperties(m))  
}
```

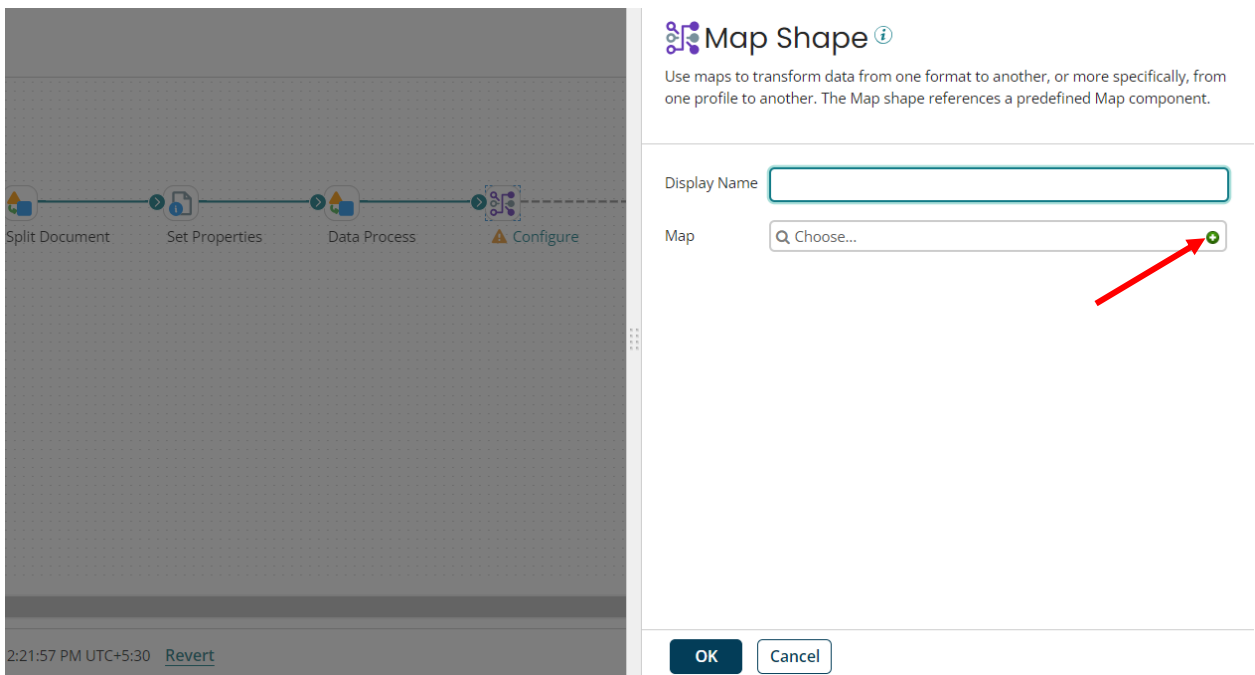
Step 27. Click on “Ok”.

- As we split the document before, now we need to combine them again.
- We will add one more processing step in same Data Process shape.

Step 28. Click on “+” icon choose Processing Step as “Combine document”, Profile Type is “Flat File” and Headers Option is “Retain First Line as Column Header”, click on “Ok”.

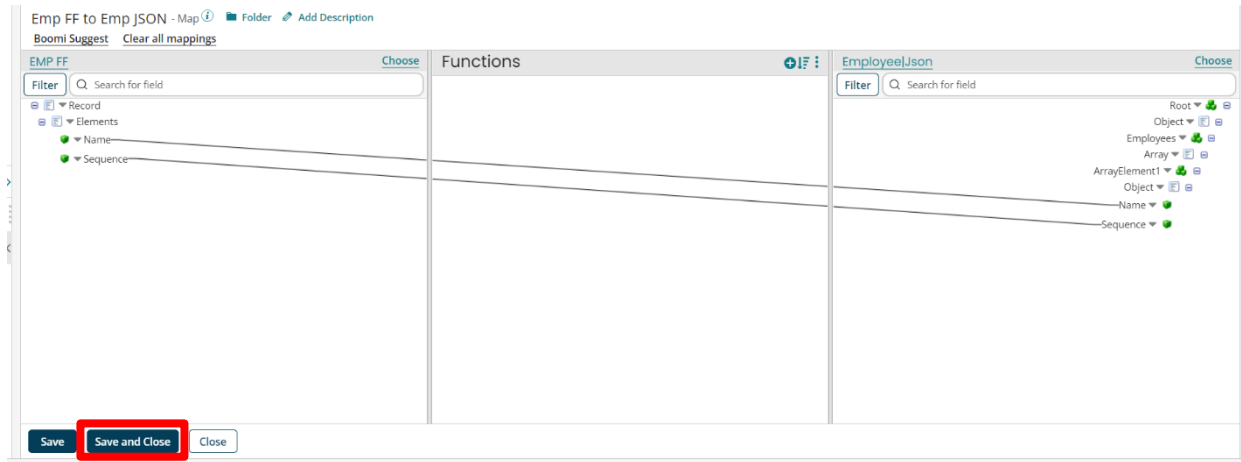


Step 29. Drag and drop a Map shape. Click on “+” icon.



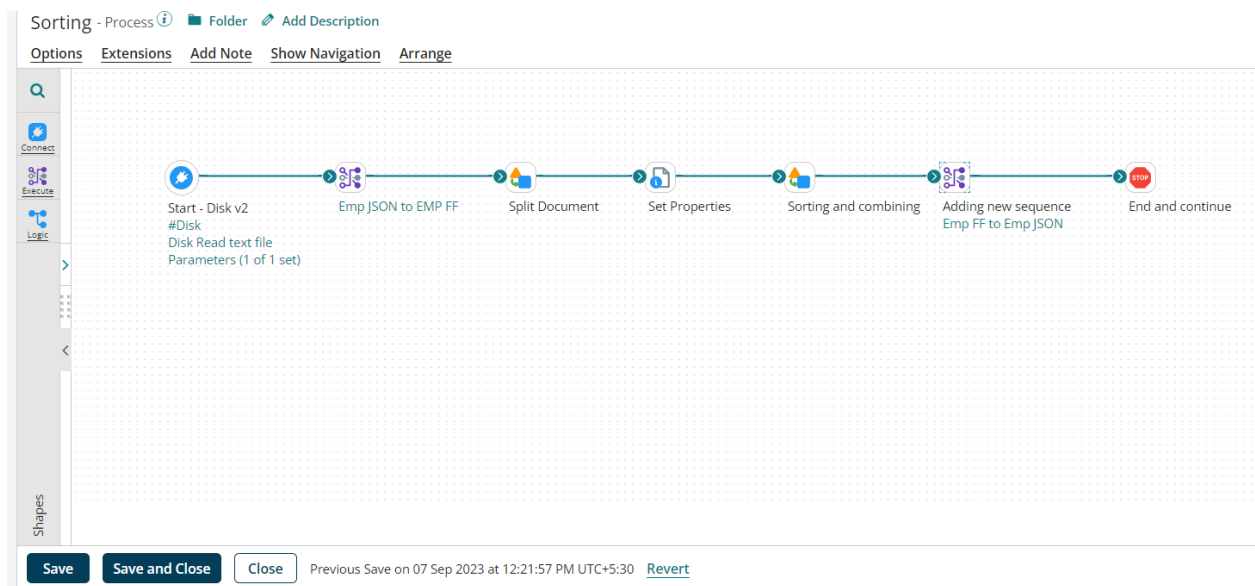
Step 30. In the place of Source choose the Emp Flat File profile which we have created previously and choose the Emp JSON profile in the Target.

Map all the fields. Click on “Save and Close”.

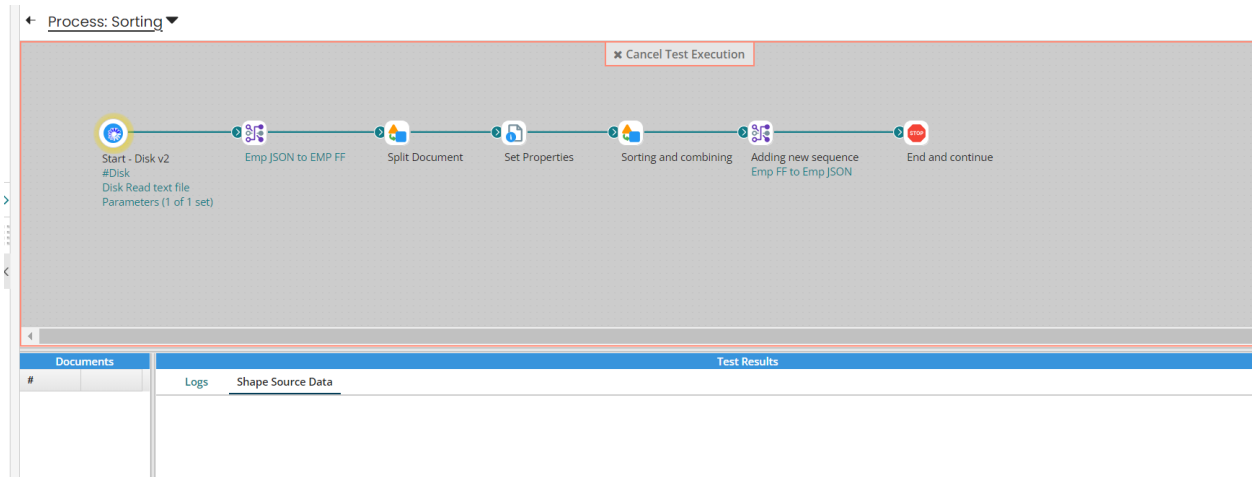


Step 31. Finally add a stop shape at the last.

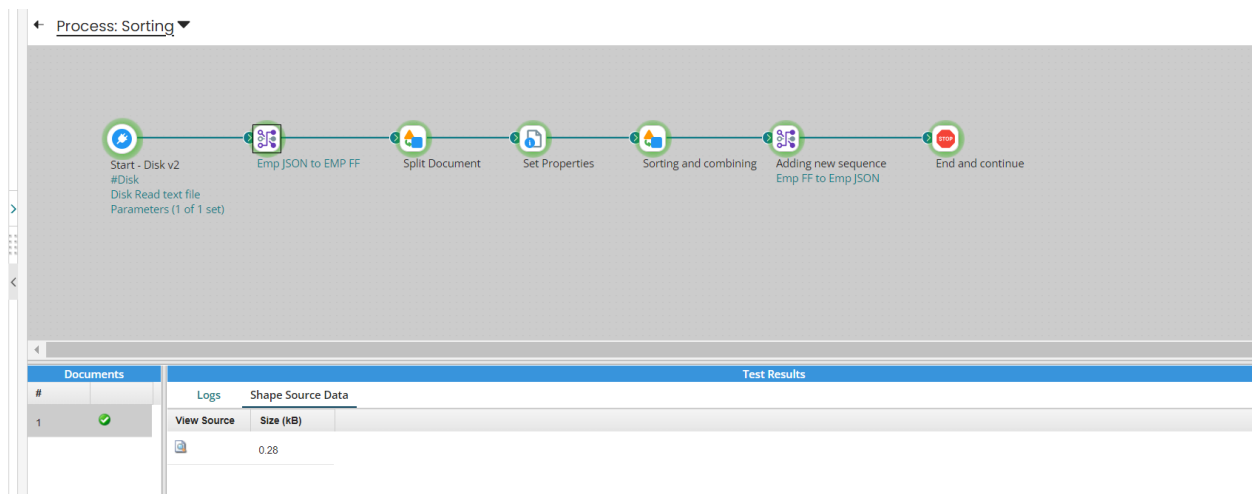
It will look like this:



Step 32. Now test the process.



Step 33. Check the document from shape source data of first Map shape. And compare with the document in the shape source data of Stop shape.





TGH

Making Integrations Simpler

boomi
Partner



TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.



Email address

connect@techygeekhub.com



Phone number

+ 011-40071137
+ 91-8810610395



Our offices

Noida Office

iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office

Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081

