



TGH

Making Integrations Simpler



Externalizing DataWeave Transformations in MuleSoft

Author

Teja Dannina



Externalizing DataWeave Transformations in MuleSoft

Overview: -

- Externalizing DataWeave transformations in MuleSoft involves separating DataWeave scripts from the main Mule configuration files to enhance organization, reusability, and maintenance.
- By externalizing these transformations, developers can manage them independently, promote code reuse, and facilitate better collaboration among team members.
- This approach streamlines development efforts, improves code maintainability, and enhances the overall efficiency of MuleSoft integration projects.

What is DataWeave: -

- DataWeave is a powerful transformation language used to manipulate and transform data within Mule applications.
- It is tightly integrated with the Mule runtime engine, which runs the scripts and expressions in your Mule app.
- DataWeave can transform data between different formats such as JSON, XML, CSV, and more.
- It enables developers to manipulate and modify data structures, including arrays, objects, and strings.
- DataWeave plays a critical role in data integration and enables MuleSoft developers to efficiently transform data as it moves through their integration workflows.

Why do we need to externalize: -

- Externalizing DWL files in Mulesoft is an additional feature to re-use the Dataweave scripts specifically.
- DWL files, or DataWeave Language files, are files that contain DataWeave code used for data transformation within MuleSoft applications. These files typically have a .dwl extension.
- DWL files are often used to encapsulate reusable data transformations.

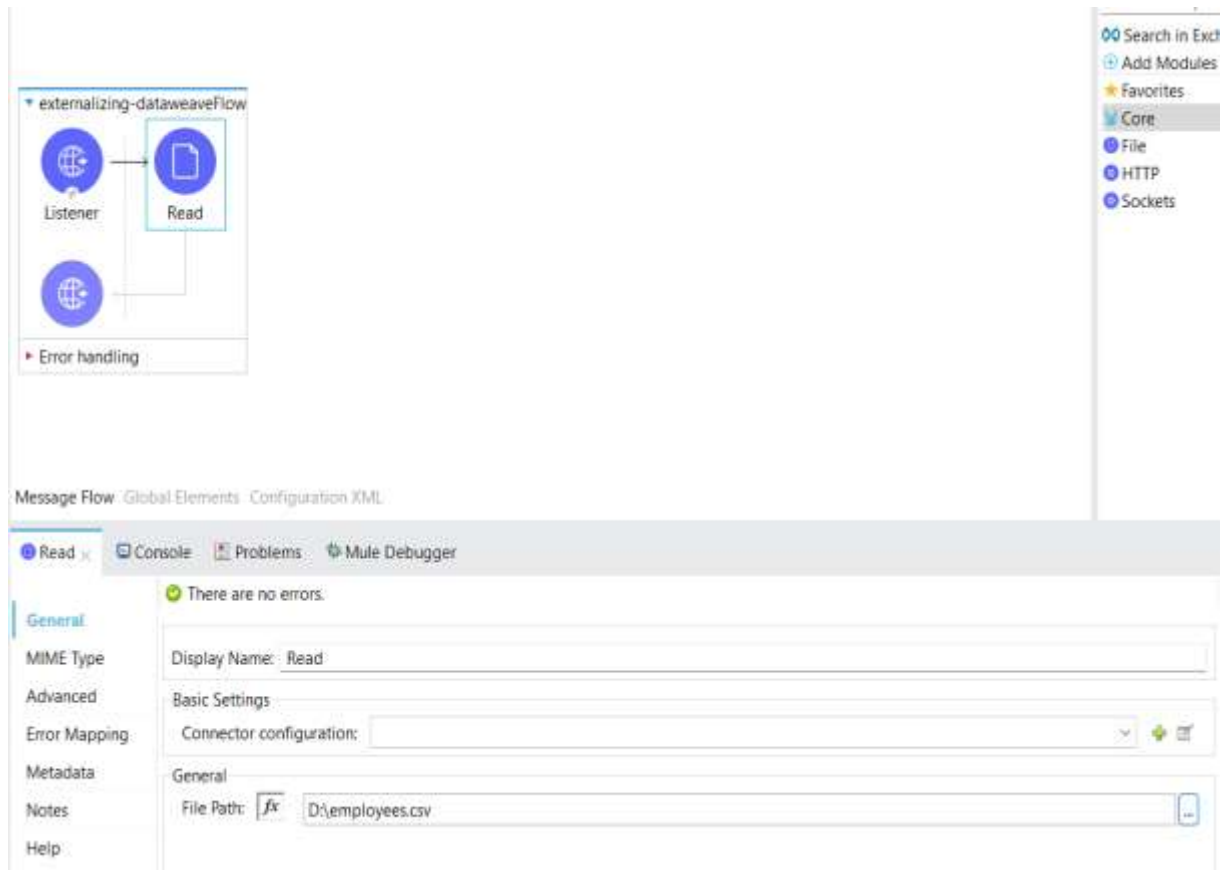
- By externalizing DataWeave script, the same script can be used at multiple places without code duplication.
- It can also make it easier to maintain all the DWL scripts used in the current project in one place in case of any errors.
- By externalizing DataWeave script, its maintainability increases as a uniform structure is followed and it was the best practice. And it was the best practice recommended by MuleSoft.

How to externalize: -

- Let's say you have a Mule application that processes customer data. The data comes in various formats (JSON, XML, CSV, etc.) and needs to be transformed into a standard format (JSON) for further processing.
- Instead of writing the transformation logic inline within each flow, you can externalize the DataWeave script into a separate .dwl file. This way, the same transformation logic can be reused across multiple flows, reducing code duplication and improving maintainability.

Steps:

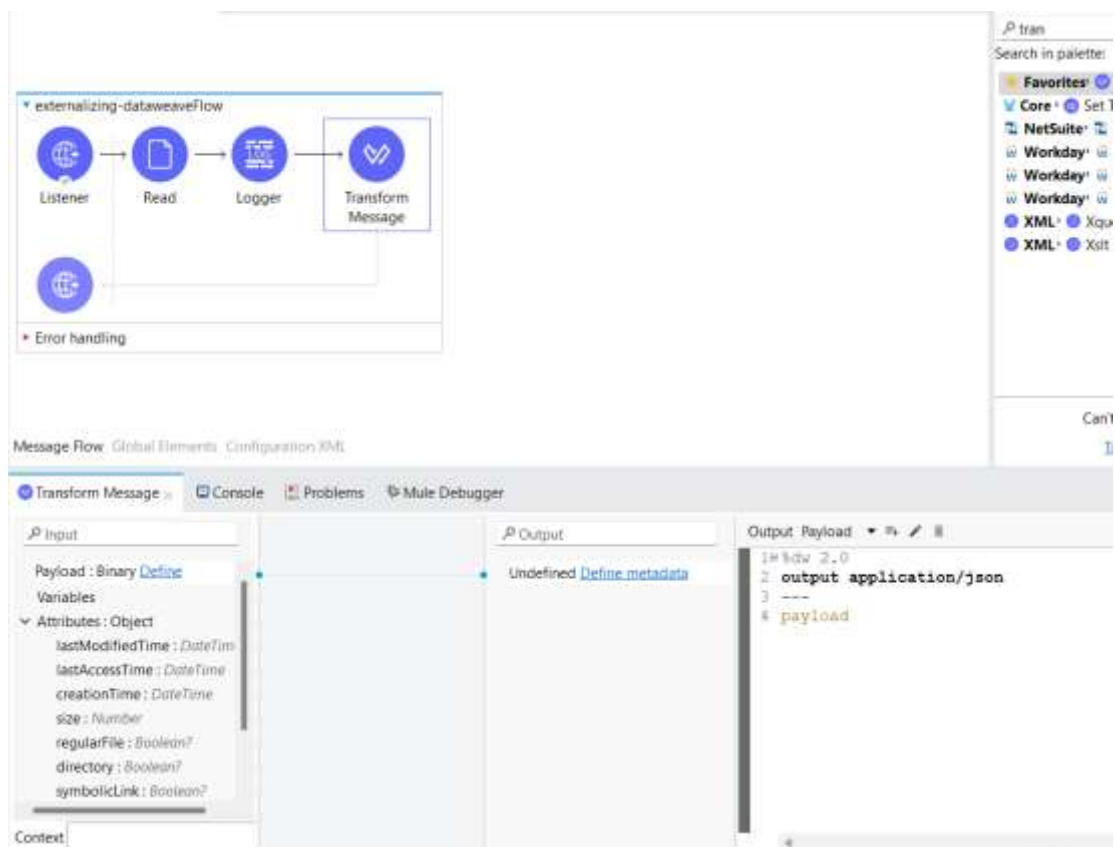
- Create a new project in AnypointStudio. To create click on the "File" menu at the top left corner of the window. From the dropdown menu, select "New" and then choose "Mule Project".
- A dialog box will appear where you can enter details about your new Mule project. Provide a name for your project in the "Project name" field. And click on finish.
- Drag and drop HTTP Listener Connector and configure its properties such as host, port, and path, and give allowed methods as POST in the advanced section and save your changes.
- Drag a Read component from the file module and drop it on the canvas
- Configure the Read component by providing file path (here I am reading a CSV file)



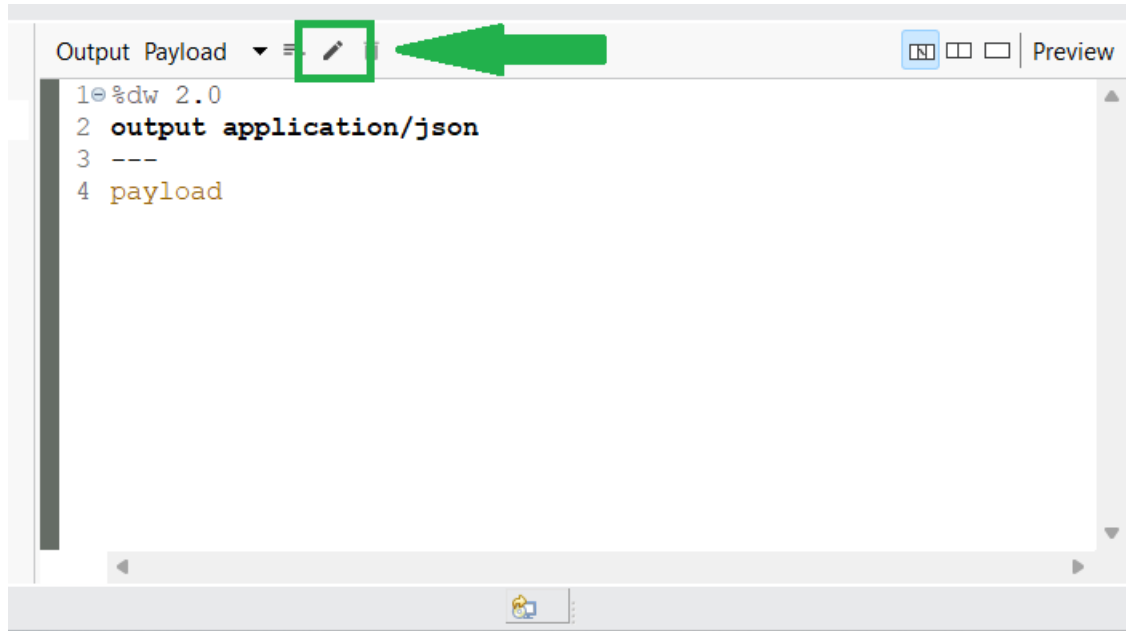
- Drag and drop the logger to log the payload from the Read component.
- Create a new folder under src/main/resources to store all the .dwl files. To create a new folder Right-click on the src/main/resources directory to open the context menu. From the context menu, select "New" and then choose "Folder". A dialog box will appear prompting you to enter the name for the new folder. Enter the desired name for your new folder and click "Finish".

- externalizing-dataweave
 - src/main/mule (Flows)
 - externalizing-dataweave.xml
 - src/main/java
 - src/main/resources
 - log4j2.xml
 - api
 - dwl
 - src/test/java
 - src/test/resources
 - src/test/munit
 - File [v1.5.1]
 - HTTP [v1.8.0]
 - JRE System Library [JDK 8 (Embedded)]
 - Mule Server 4.5.2 EE
 - Sockets [v1.2.3]
 - exchange-docs
 - src
 - target
 - mule-artifact.json
 - pom.xml [Mule Server 4.5.2 EE]

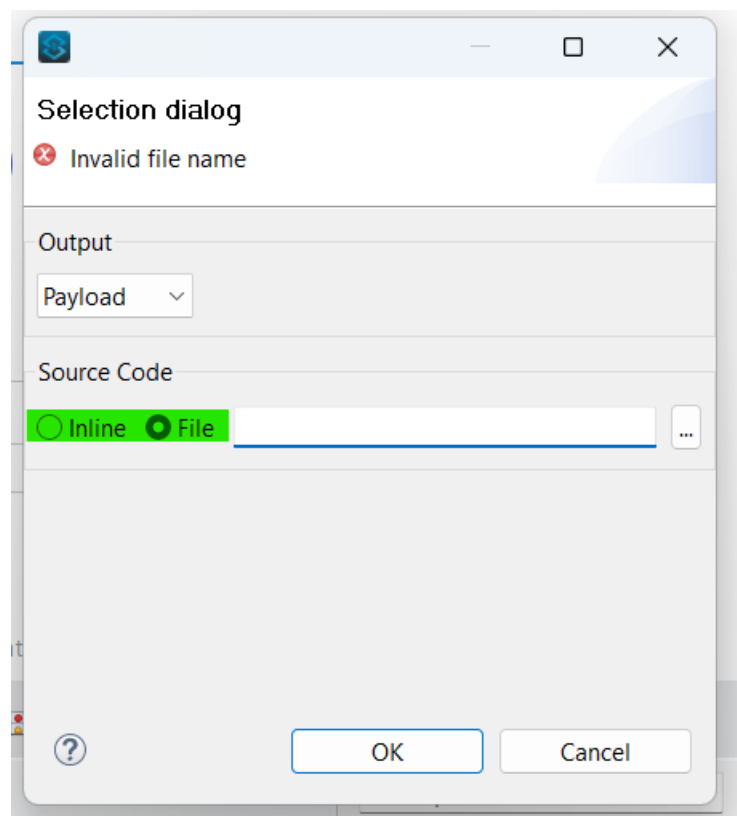
- It is recommended to have a folder just dedicated to .dwl files for easier access to files.
- Drag a Transform Message and drop it on the canvas.



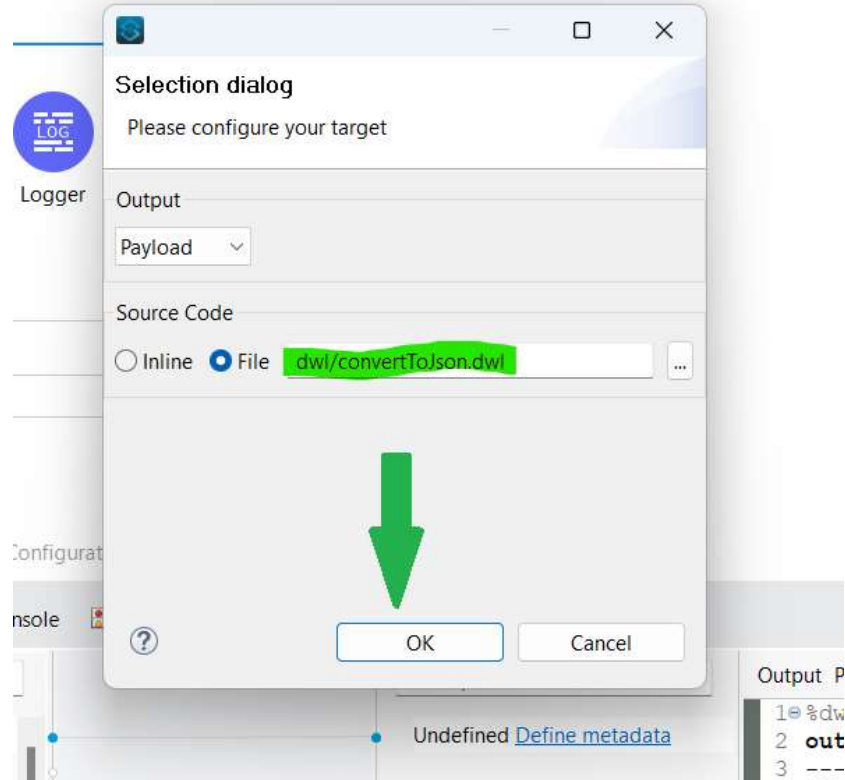
- Write a DataWeave code as the transform message (here I am converting the payload to JSON)
- To externalize, Click on the edit icon as shown in the image below.



- Once you click, you'll get a pop-up dialog window. In the dialog box, by default, the source code is selected to Inline. Change the source code to the File option.



- It enables the filename section to refer to the respective .dwl file to be used once you select the File option. If you have the destination file path you can directly paste the path. Or you can click 3 dots which open a file explorer to point to your file and then click 'OK'.



- We can check that a new .dwl file has been created under dwl folder.
- This is how the Configuration XML looks when you refer to a file.

```
<ee:transform doc:name="Transform Message" doc:id="72dcd49-702e-4a59-a36b-14eb5808178b" >
  <ee:message >
    <ee:set-payload resource="dwl/convertToJson.dwl" />
  </ee:message>
</ee:transform>
```
- Upon executing our application and submitting a request, we observe the transformation of CSV data into JSON format.
- Please be aware that any modifications made to the DWL file will affect all transformations, so exercise caution before proceeding with changes; it's advisable to externalize all mappings as DWL files to prevent code duplication and adhere to best practices.



TGH

Making Integrations Simpler

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

