

# Troubleshooting Data Preview issues

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

## Contents

- [Issue](#)
- [Investigation](#)
- [Debug Sampling](#)
- [Other Issues/Pre-checks](#)

## Issue

Sometimes customers get error during data preview, or validating schema or missing data. In such cases, you can troubleshoot those issues by capturing the [browser trace](#) and [Pipeline support files](#) for further investigation.

NOTE: If the issue is related to schema related or missing columns, try importing the schema or import projection as per this [TSG](#) and retry.

## Investigation

If you encounter any issues in Dataflow Preview, capture the [browser trace](#), and check the **executeDataFlowDebugCommand** and **dfCommandOutput** responses:

The screenshot displays two network traces from a browser's developer tools. The top trace is named 'executeDataFlowDebugCommand?api-version=2018-06-01' and is highlighted with a red box. Its preview response shows a JSON object with a 'runId' and a 'runId' field. The bottom trace is named 'dfCommandOutput\$e01604f2-232e-42bd-b35c-8ec8e' and is also highlighted with a red box. Its preview response shows a JSON object with a 'status' field set to 'Succeeded' and a 'data' field containing an 'output' string with various statistics like 'rowCount', 'errorCount', 'updateCount', 'deleteCount', 'partitionCounts', and 'columnStatistics'.

The data flow code is executed on the backend databricks cluster, and the response is sent to the UX. You would notice the JSON containing the data (response) from the cluster, which is previewed in the Data Preview tab in the dfCommandOutput output. If there's an error from the backend cluster, it would be shown in the dfCommandOutput output as well.

You would notice executeDataFlowDebugCommand called twice whenever you click on Data Preview. Basically, dataflow-debug-client-service calls the executePreviewQuery and executeStatisticsQuery requests to the backend service to process the request. Refer to the following table for more details:

| API Call                  | Description  |
|---------------------------|--|
| addDataFlowToDebugSession | Submits the dsl to the dataflow session for execution. If the job is failing to submit to the dataflow debug session, use <b>HttpIncomingRequests</b> and <b>HttpOutgoingRequests</b> tables of Armprod database to check for any failures. Eg. <code>```HttpIncomingRequests</code> |
| executePreviewQuery       | Shows the sample data in the Data Preview tab.   |
| executeStatisticsQuery    | Shows the statistics related to the Data like rowCount, errorCount, updateCount, deleteCount, columnStatistics etc.  |

Check the request payload of the HTTP POST request to find out the function called for execution by executeDataFlowDebugCommand and current **sessionId**.

The screenshot shows a REST client interface with a list of API calls on the left. The selected call is `executeDataFlowDebugCommand?api-version=2018-06-01`. The right pane shows the request details for this call. Under the 'Request Payload' section, the JSON body is displayed:

```
{
  "commandPayload": {
    "streamName": "Movies",
    "rowLimits": 100,
    "schemaOnly": true
  },
  "command": "executePreviewQuery",
  "commandPayload": {
    "streamName": "Movies",
    "rowLimits": 100,
    "schemaOnly": true
  },
  "dataFlowName": "MovieDemo",
  "sessionId": "3b1eb222-c369-48c1-a890-c4576d702ad2",
  "userSessionId": "86f2ca62-a6d0-439d-8878-ca024d4b9db6"
}
```

Query the query **DataflowDebugEvent** table to check the various API requests and status.

```
cluster('adfcus').database('AzureDataFactory').DataflowDebugEvent | union cluster('adfneu').database('AzureDataFactory').DataflowDebugEvent
| where PreciseTimeStamp >= ago(1h)
| where runId == "<sessionId>"
| project PreciseTimeStamp, Role, operationName, state, runId, dataflowSessionId, traceCorrelationId
```

If there's a failure in any of the above request/response, with the **traceCorrelationId** value, you may further investigate the issue in the **ApiOperationEvent** or **AdfTraceEvent** table.

## Debug Sampling

Please refer to this [Wiki](#)

## Other Issues/Pre-checks

If the data preview is still not showing some values or nulls, you may also check the following:

1. If the schema is imported for the source data in dataflow, and see if all the columns are showing their datatypes.
2. If the option to schema drift is set to true or not in the source. If it's set to false, try setting to true and do a schema import.
3. If the multiline-row in the source is set as true. If the data in each row in source is not multiline, set this option to false and retry.
4. If the rowdelimiter in the source file is `\r\n` then set the rowdelimiter as `\n` and retry.

### Additional Information:

- **Icm References:**
- **Author:** vimals
- **Reviewer:** vimals
- **Keywords:**

### How good have you found this content?

