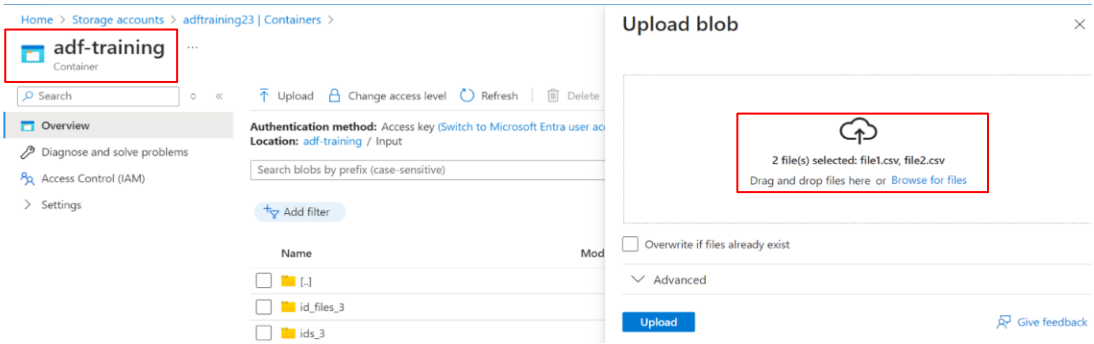
<u>Formal Documentation of Azure Data Factory Pipeline – Training</u>

<u>Use Case: Unpivoting the File</u>

1. Uploading Input File in the Container

- A container named adf-training was previously created in the Azure storage account.
- For this pipeline, the file was uploaded into the container in the input folder.
- Once the upload was successful, the file was added to the input folder.
- Files uploaded: file1.csv, file2.csv (refer image 1).
- We had 2 source files: one containing the departments table and the other containing only the ids. The task was to unpivot the department file and compare the IDs file with the department file. If the id is available in the department file, then load the data to the destination.


Home > Storage accounts > adftraining23 | Containers >
adf-training
Container
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Upload
Change access level
Refresh
Delete
Search blobs by prefix (case-sensitive)
Add filter
Name
Mod
[..]
id_files_3
ids_3
Upload blob
Drag and drop files here or Browse for files
Overwrite if files already exist
Advanced
Upload
Give feedback

Authentication method: Access key (Switch to Microsoft Entra user ac
Location: adf-training / Input

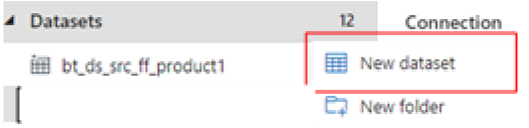
2 file(s) selected: file1.csv, file2.csv

-Image

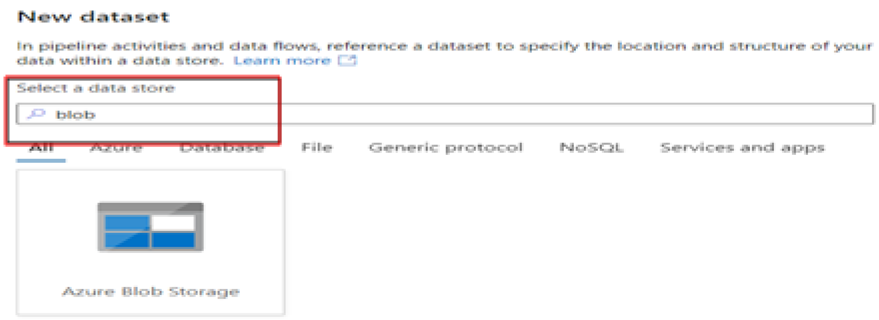
2. Dataset Creation.

<u>Source dataset 1</u>

- A new source dataset was created for the source data file.
- Source dataset name: bt_ds_src_dpt_transformation
- Inside Azure Data Factory, in the Author tab, I selected the Dataset option and clicked on "New Dataset" (refer to image 1). I chose the Azure Blob Storage option (refer to image 2).
- Next, I selected the Delimited Text file format, which brought me to the properties page where I defined the dataset name and path (refer to image 3).
- I specified the dataset name, selected the linked service, and provided the path of my input file.
- These steps created my source dataset that contained department information (refer image 4).



- Image 1

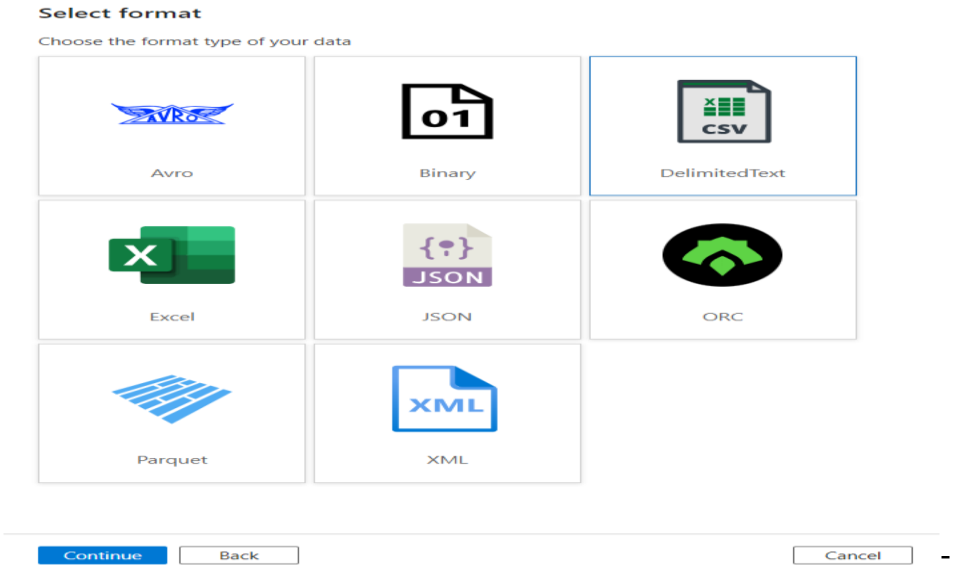


New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store: blob

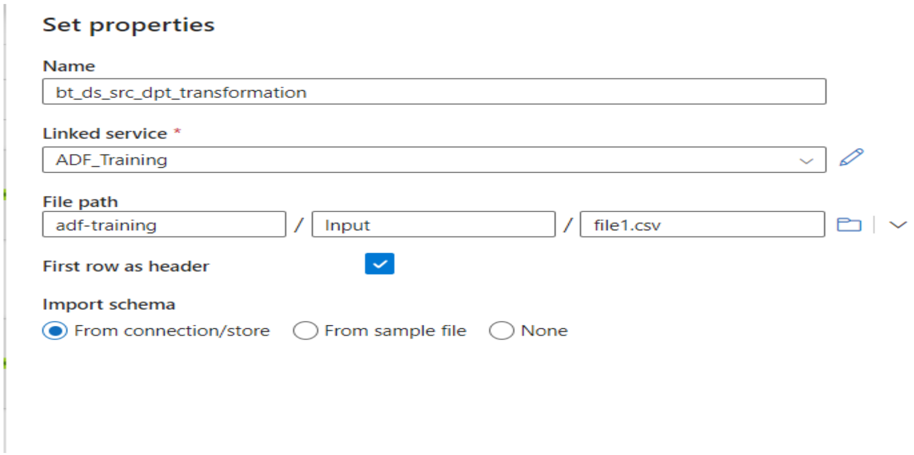
- Image 2



Select format

Choose the format type of your data

-Image 3



Set properties

Name: bt_ds_src_dpt_transformation

Linked service *: ADF_Training

File path: adf-training / Input / file1.csv

First row as header: ☑

Import schema: ● From connection/store ○ From sample file ○ None

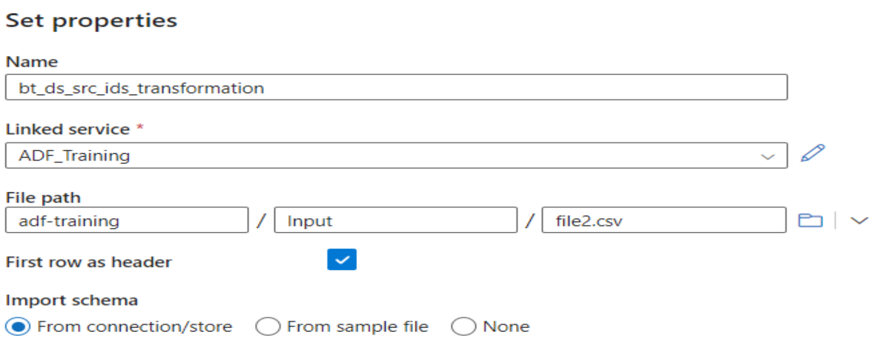
-Image 4

Source dataset 2

Source dataset 2

- A new source dataset was created for the source2 data file.
- Source dataset name: bt_ds_src_ids_transformation

- I followed similar steps for source dataset 2.
- In the Author tab, I selected the datasets, clicked on "New Dataset," selected Azure Blob Storage, and then selected the Delimited Text format, which brought me to the properties page where I defined the dataset name and path. These steps created my source dataset that contained IDs information (refer image 1).



Set properties

Name: bt_ds_src_ids_transformation

Linked service *: ADF_Training

File path: adf-training / Input / file2.csv

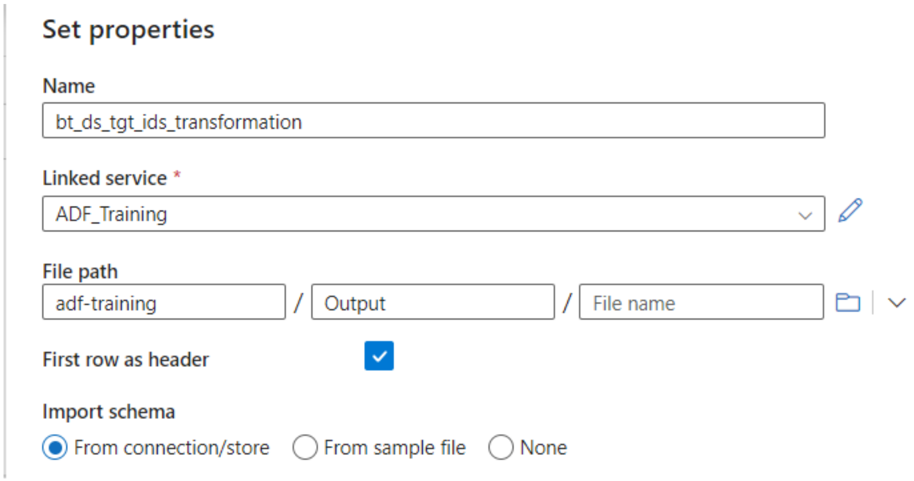
First row as header: ☑

Import schema: ● From connection/store ○ From sample file ○ None

-Image 1

Target dataset

- A new source dataset was created for the target data file.
- Source dataset name: bt_ds_tgt_ids_transformation
- I followed similar steps for the target dataset. In the Author tab, I selected the datasets, clicked on "New Dataset," selected Azure Blob Storage, and then selected the Delimited Text format, which brought me to the properties page where I defined the dataset name and path (refer image 1)



Set properties

Name: bt_ds_tgt_ids_transformation

Linked service *: ADF_Training

File path: adf-training / Output / File name

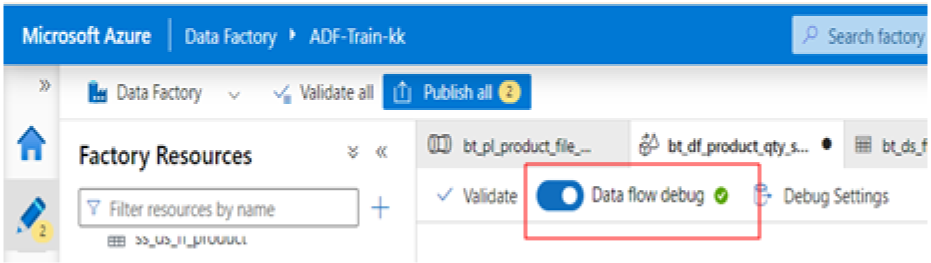
First row as header: ☑

Import schema: ● From connection/store ○ From sample file ○ None

- Image 1

3. Dataflow Creation

- Dataflow name: bt_df_file_transformation_unpivot.
- The mandatory step is to enable dataflow debug so we can preview the data at every step (refer to image 1).

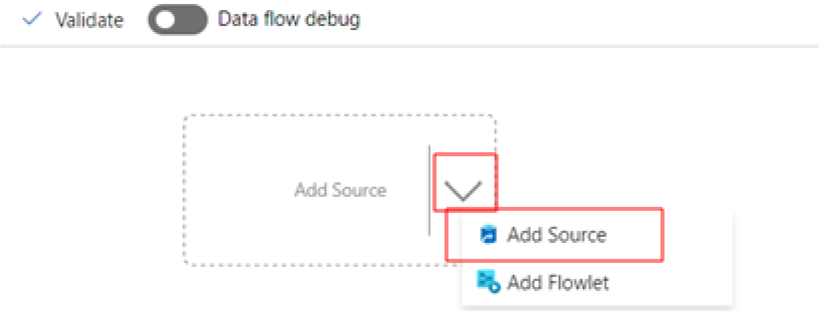


-Image 1

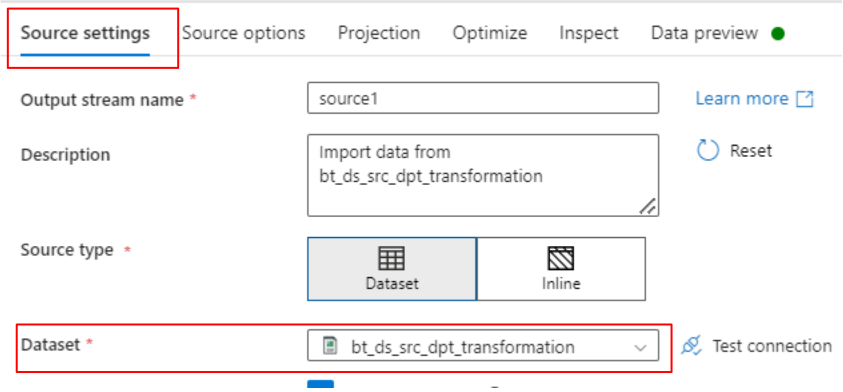
<u>Dataflow Steps:</u>

A. Selecting Source

- Added the sources to add my 2 seperate source file (refer to image 1).
- In source settings, I added my source dataset (refer to image 2,3).



-Image 1



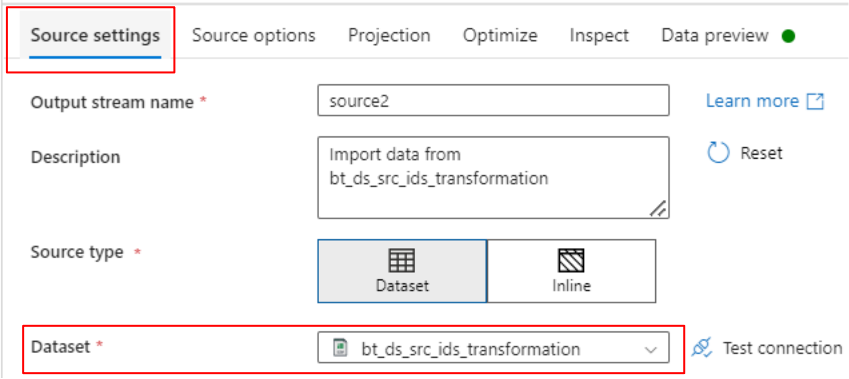
Output stream name * source1

Description: Import data from bt_ds_src_dpt_transformation

Source type * Dataset | Inline

Dataset * bt_ds_src_dpt_transformation

-Image 2



Source settings | Source options | Projection | Optimize | Inspect | Data preview ●

Output stream name * source2

Description: Import data from bt_ds_src_ids_transformation

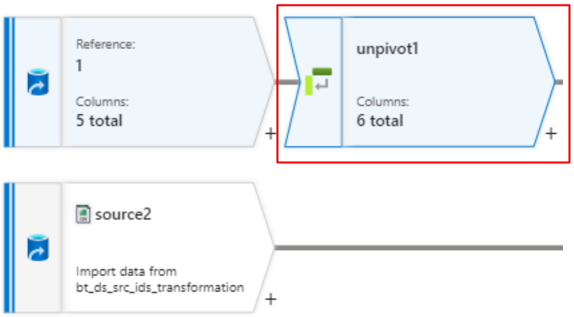
Source type * Dataset | Inline

Dataset * bt_ds_src_ids_transformation

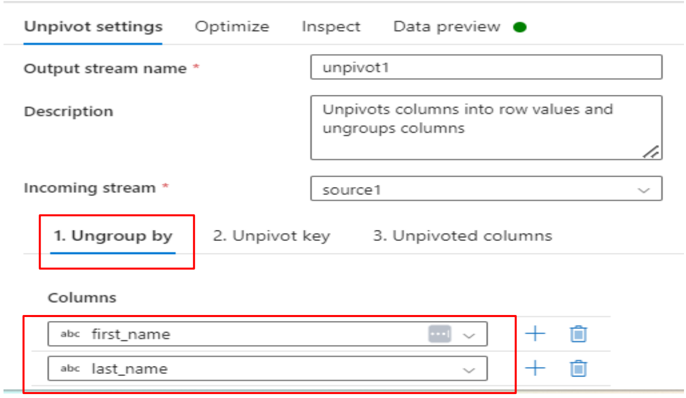
-Image 3

B. Unpivot

- Click on the "Add transformation (+)" button and select "Unpivot" (Image 1).
- The unpivot function in Azure Data Factory's Data Flow transforms columns into rows, converting wide data formats into a long, normalized format by moving the values of selected columns into rows under a new column.
- The ungroup by option in the Unpivot transformation allows you to specify columns that should not be unpivoted. I selected the columns first_name, last_name (refer image 2).
- Then there is the unpivot key, which specifies the name of the new column that will hold the original column names. I selected the ids column with a data type of integer (refer image 3).
- Moved to unpivoted columns, the Unpivot transformation allows you to convert columns into rows, making your data normalized from wide to long format. I selected the departments columns that I wanted to unpivot: dpt_id, mkt_id, fin_id (refer image 4).



-Image 1



Unpivot settings | Optimize | Inspect | Data preview ●

Output stream name * unpivot1

Description: Unpivots columns into row values and ungroups columns

Incoming stream * source1

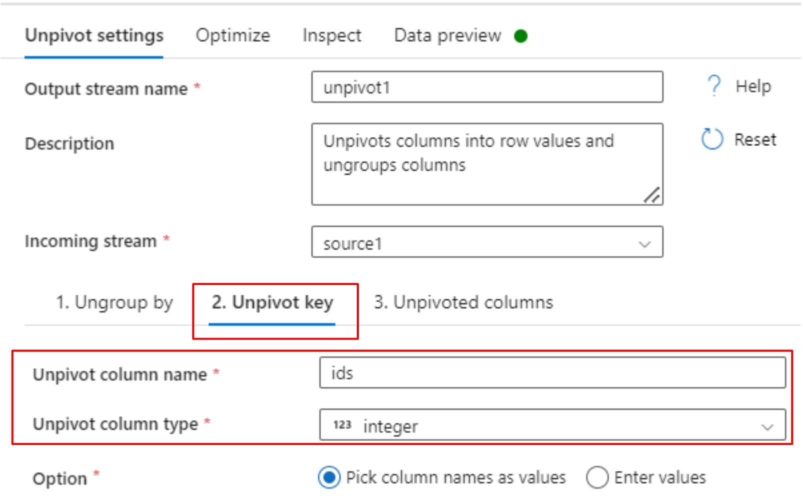
1. Ungroup by | 2. Unpivot key | 3. Unpivoted columns

Columns

abc first_name

abc last_name

-Image 2



Unpivot settings | Optimize | Inspect | Data preview ●

Output stream name *: unpivot1

Description: Unpivots columns into row values and ungroups columns

Incoming stream *: source1

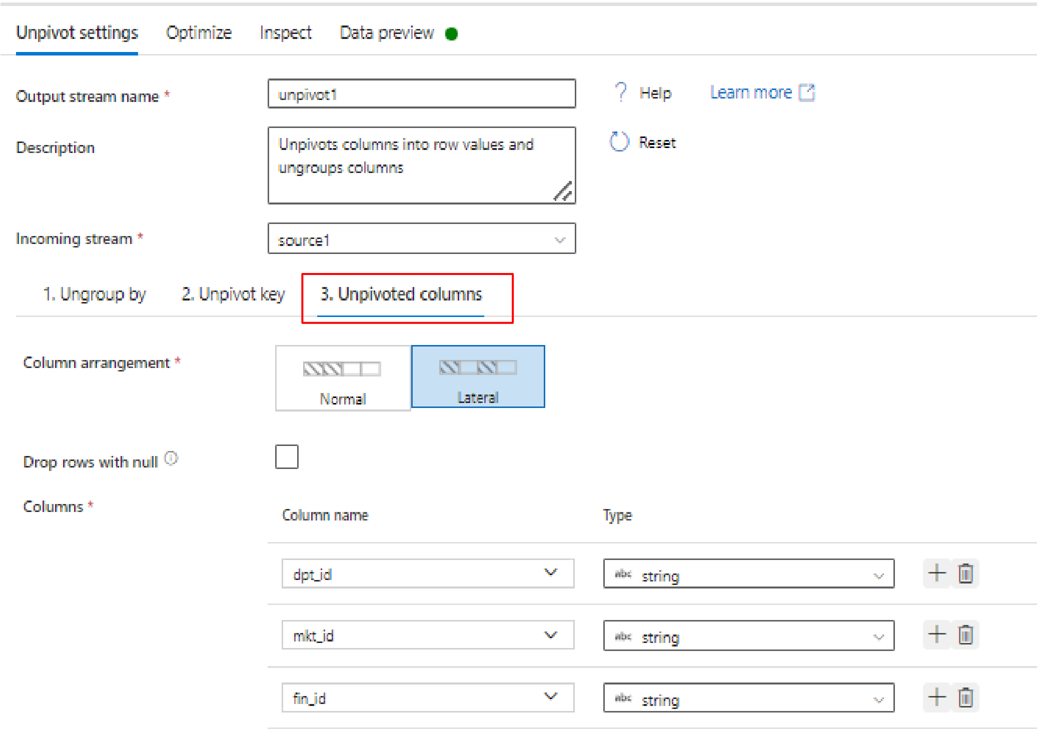
1. Ungroup by | 2. Unpivot key | 3. Unpivoted columns

Unpivot column name *: ids

Unpivot column type *: 123 integer

Option *: ● Pick column names as values ○ Enter values

-Image 3



Unpivot settings | Optimize | Inspect | Data preview ●

Output stream name *: unpivot1

Description: Unpivots columns into row values and ungroups columns

Incoming stream *: source1

1. Ungroup by | 2. Unpivot key | 3. Unpivoted columns

Column arrangement *: Normal | Lateral

Drop rows with null: □

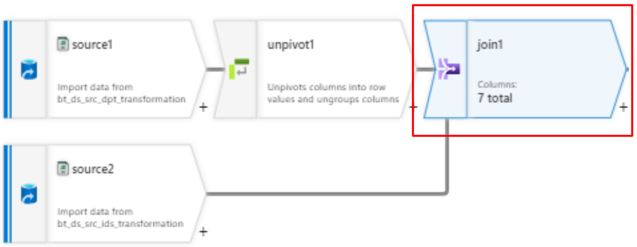
Columns *

Column name	Type
dpt_id	abc string
mkt_id	abc string
fin_id	abc string

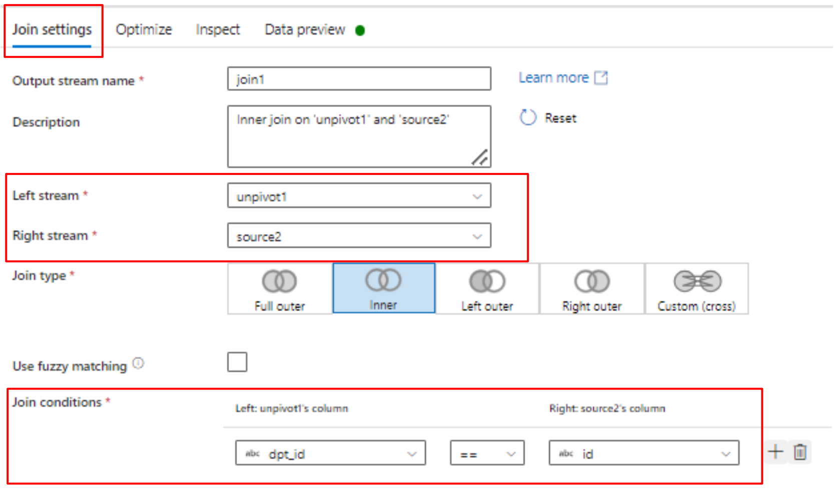
-Image 4

C. Join

- Click on the "Add transformation (+)" button and select "Join" (refer image 1).
- I used the join function to join the Unpivot transformation with source 2. Selected the left stream as Unpivot1 and right stream as source 2. Join type is "Inner Join".
- Mentioned the joining conditions (refer image 2).


source1
Import data from bt_ds_src_dpt_transformation
unpivot1
Unpivots columns into row values and ungroups columns
join1
Columns:
7 total
source2
Import data from bt_ds_src_ids_transformation

-Image 1



Output stream name * join1

Description: Inner join on 'unpivot1' and 'source2'

Left stream * unpivot1

Right stream * source2

Join type * Full outer / [✓ Inner] / Left outer / Right outer / Custom (cross)

Use fuzzy matching □

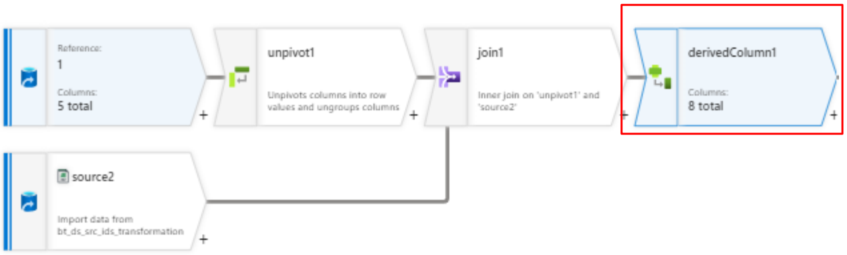
Join conditions *

Left: unpivot1's column		Right: source2's column
abc dpt_id	==	abc id

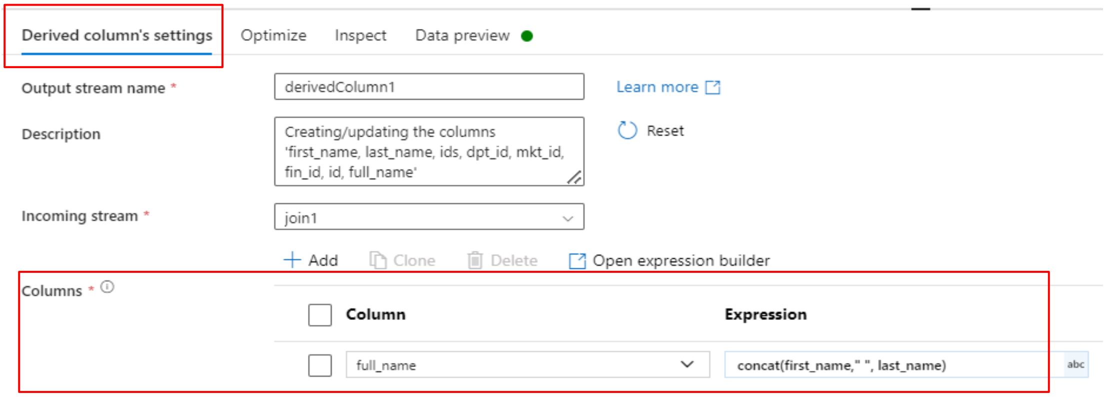
-Image 2

D. Derived Column

- Click on the "Add transformation (+)" button and select "Derived Column" (refer image 1).
- Selected this function to concatenate the first and last names.
- In the columns option, assigned a column name as"full_name" and wrote a concat expression (refer image 2).


Reference:
1
Columns:
5 total
unpivot1
Unpivots columns into row values and ungroups columns
join1
Inner join on 'unpivot1' and 'source2'
derivedColumn1
Columns:
8 total
source2
Import data from bt_ds_src_ids_transformation

-Image 1



Derived column's settings | Optimize | Inspect | Data preview ●

Output stream name * : derivedColumn1 — Learn more

Description: Creating/updating the columns 'first_name, last_name, ids, dpt_id, mkt_id, fin_id, id, full_name' — Reset

Incoming stream * : join1

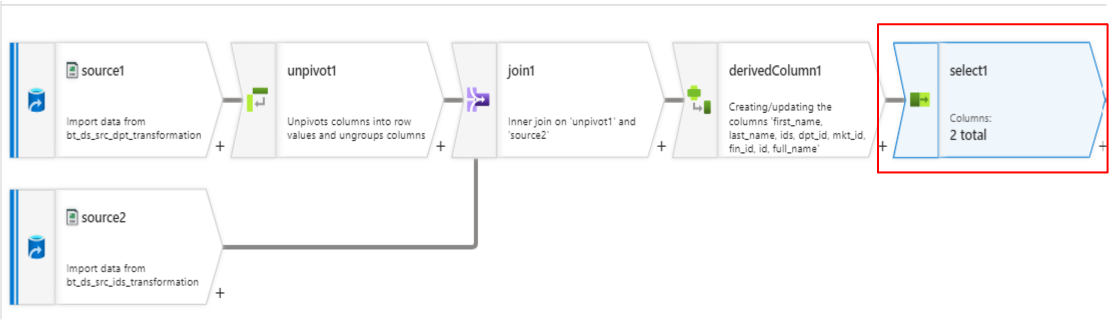
Columns * — + Add | Clone | Delete | Open expression builder

	Column	Expression
□	full_name	concat(first_name," ", last_name)

-Image 2

E. Select

- Click on the "Add transformation (+)" button and select "Select" (refer image 1).
- After the derived column function, we had 8 columns. I used the select function to select only the necessary columns, selecting the dpt_id, full_name column (refer image 2).



source1 — Import data from bt_ds_src_dpt_transformation

unpivot1 — Unpivots columns into row values and ungroups columns

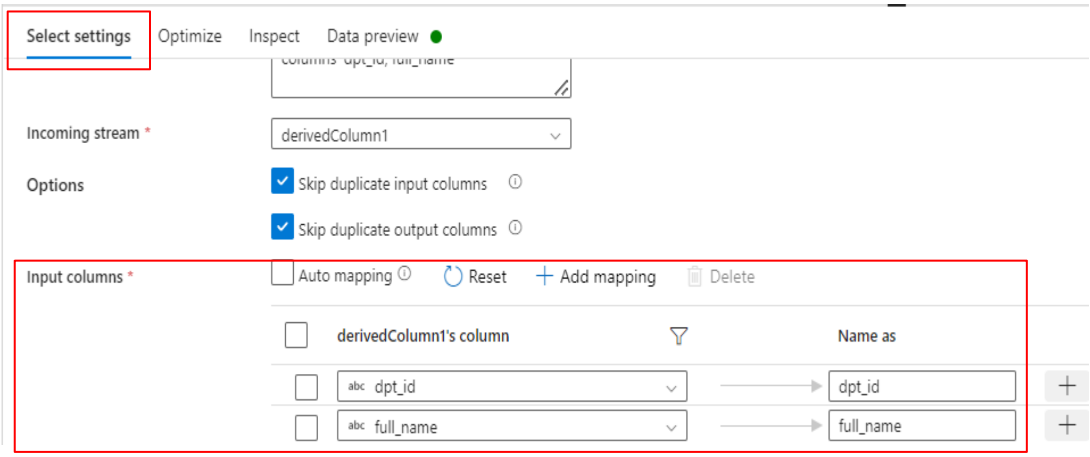
join1 — Inner join on 'unpivot1' and 'source2'

derivedColumn1 — Creating/updating the columns 'first_name, last_name, ids, dpt_id, mkt_id, fin_id, id, full_name'

select1 — Columns: 2 total

source2 — Import data from bt_ds_src_ids_transformation

-Image 1



Select settings | Optimize | Inspect | Data preview ●

columns 'dpt_id, full_name

Incoming stream * : derivedColumn1

Options: ☑ Skip duplicate input columns ☑ Skip duplicate output columns

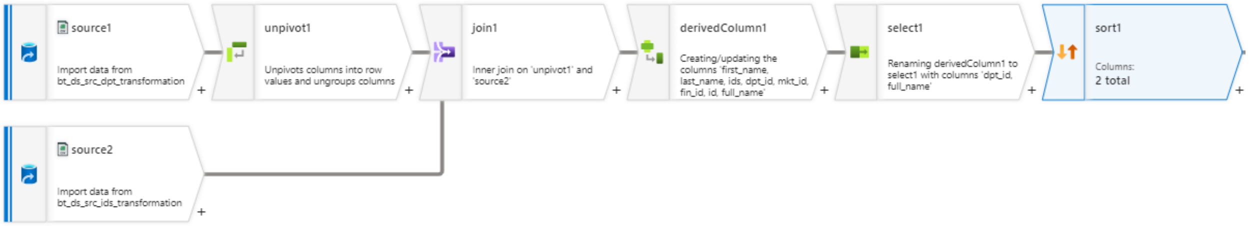
Input columns * — □ Auto mapping | Reset | + Add mapping | Delete

□	derivedColumn1's column	Name as
□	abc dpt_id	dpt_id
□	abc full_name	full_name

-Image 2

F. Sort

- Click on the "Add transformation (+)" button and select "Sort" (refer image 1).
- Added the sort function to sort the dpt_id in ascending order (refer image 2).



source1
Import data from bt_ds_src_dpt_transformation

unpivot1
Unpivots columns into row values and ungroups columns

join1
Inner join on 'unpivot1' and 'source2'

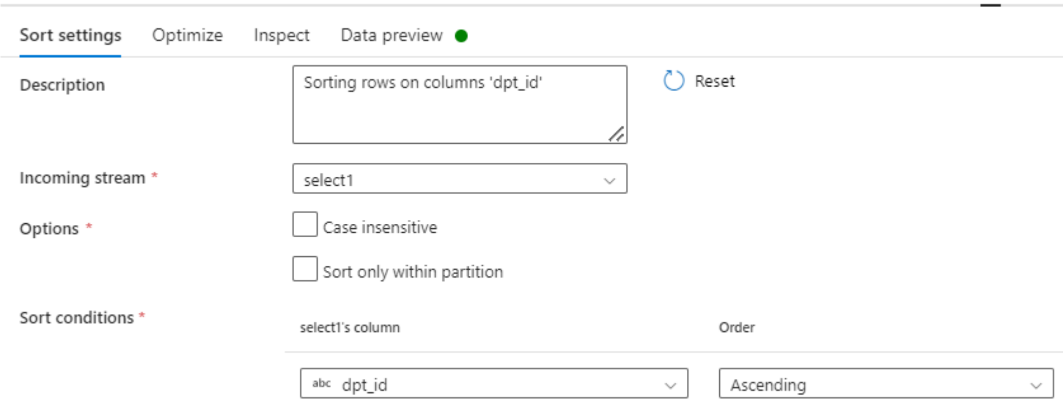
derivedColumn1
Creating/updating the columns 'first_name, last_name, ids, dpt_id, mkt_id, fin_id, id, full_name'

select1
Renaming derivedColumn1 to select1 with columns 'dpt_id, full_name'

sort1
Columns:
2 total

source2
Import data from bt_ds_src_ids_transformation

-Image 1



Sort settings | Optimize | Inspect | Data preview ●

Description: Sorting rows on columns 'dpt_id'

Reset

Incoming stream *: select1

Options *:
☐ Case insensitive
☐ Sort only within partition

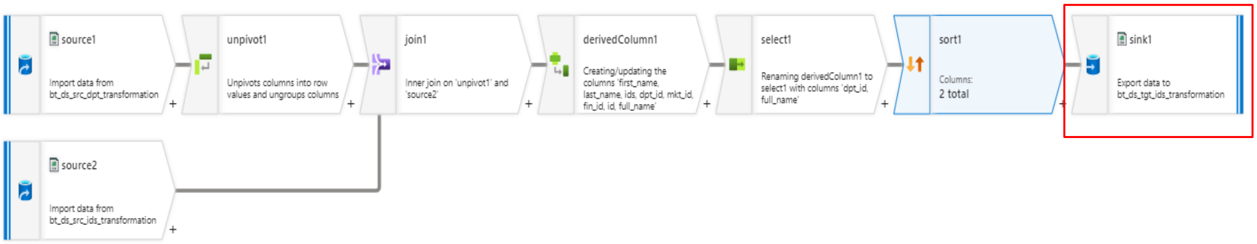
Sort conditions *

select1's column	Order
abc dpt_id	Ascending

-Image 2

G. Sink

- Click on the "Add transformation(+)" button and select "Sink" (refer image 1).
- Added my target dataset in the sink and in settings selected the file name option as Output to single file so I can assign a name to my output file (refer image 2,3).



source1
Import data from bt_ds_src_dpt_transformation

unpivot1
Unpivots columns into row values and ungroups columns

join1
Inner join on 'unpivot1' and 'source2'

derivedColumn1
Creating/updating the columns 'first_name, last_name, ids, dpt_id, mkt_id, fin_id, id, full_name'

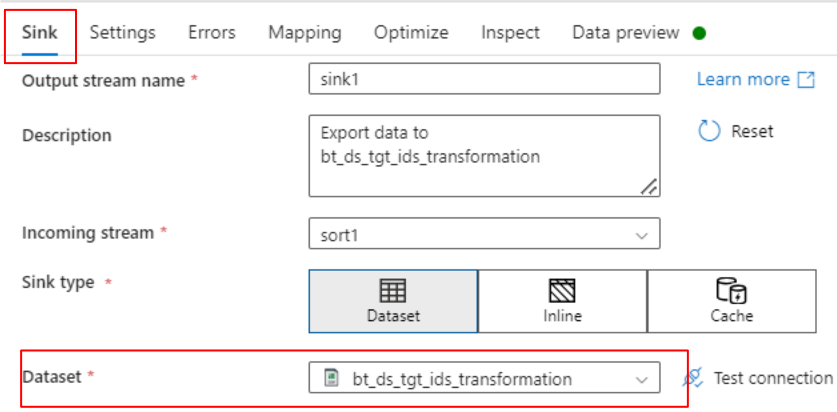
select1
Renaming derivedColumn1 to select1 with columns 'dpt_id, full_name'

sort1
Columns:
2 total

sink1
Export data to bt_ds_tgt_ids_transformation

source2
Import data from bt_ds_src_ids_transformation

-Image 1



Output stream name * sink1

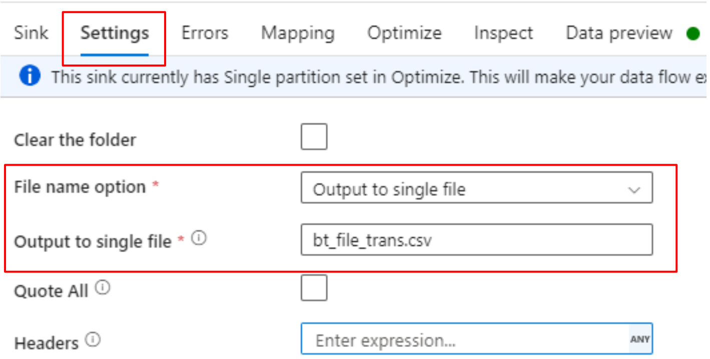
Description: Export data to bt_ds_tgt_ids_transformation

Incoming stream * sink1 sort1

Sink type * Dataset | Inline | Cache

Dataset * bt_ds_tgt_ids_transformation

-Image 2



This sink currently has Single partition set in Optimize. This will make your data flow e

Clear the folder □

File name option * Output to single file

Output to single file * bt_file_trans.csv

Quote All □

Headers: Enter expression...

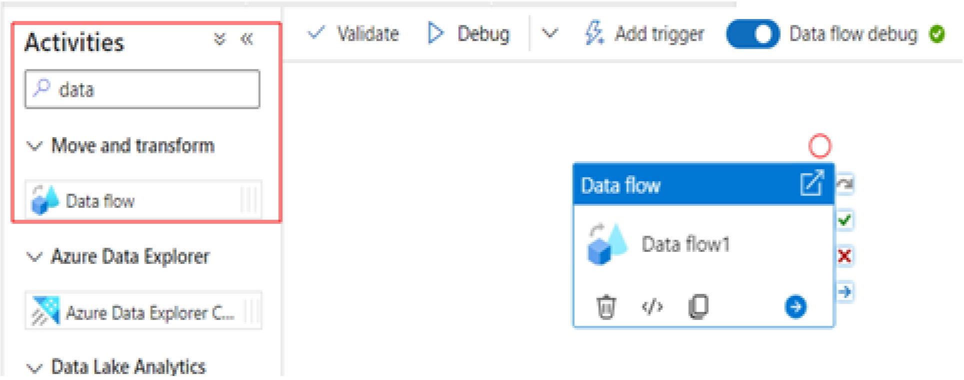
-Image 3

4. Pipeline Creation

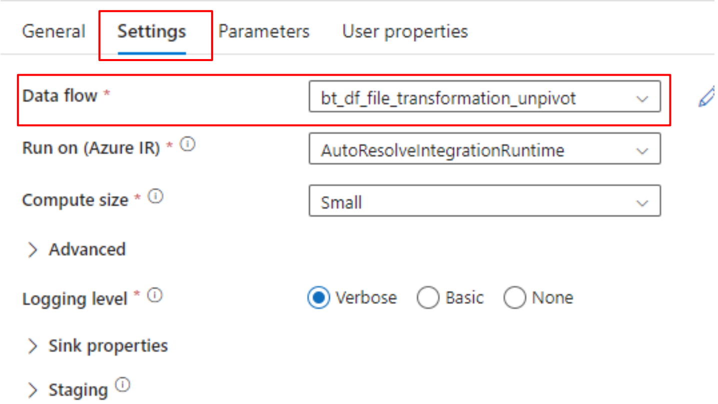
- I then created a pipeline named bt_pl_file_transformation to execute the dataflow activities.

5. Dataflow Activity

- From the activities, I dragged and dropped the dataflow activity (refer image 1).
- Then, in the settings, I select the dataflow (refer image 2).



-Image 1



General | **Settings** | Parameters | User properties

Data flow * bt_df_file_transformation_unpivot

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute size * Small

Advanced

Logging level * ● Verbose ○ Basic ○ None

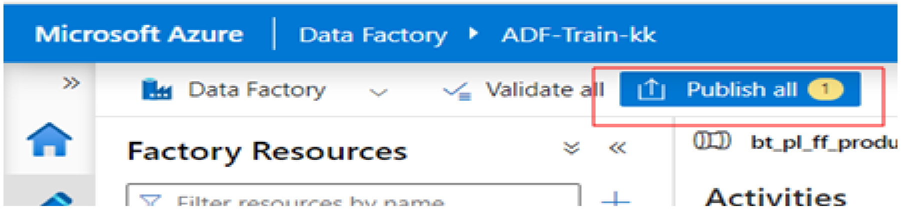
Sink properties

Staging

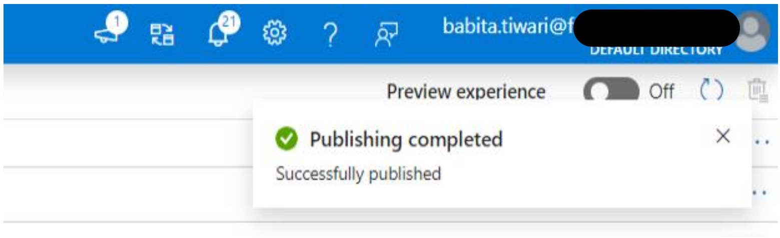
-Image 2

6. Publishing and Executing the Pipeline

- The activities were saved/published, and all were successfully published and executed via debug tab. (refer to images 1, 2, 3).



- Image 1



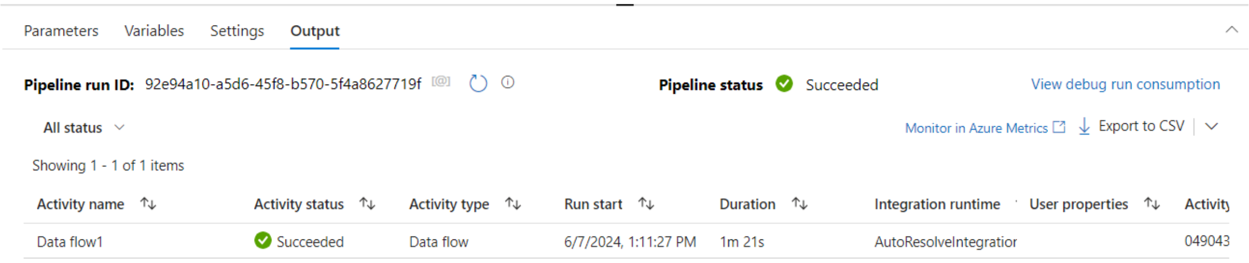
babita.tiwari@f[redacted]

Preview experience Off

Publishing completed

Successfully published

-Image 2



Parameters | Variables | Settings | **Output**

Pipeline run ID: 92e94a10-a5d6-45f8-b570-5f4a8627719f

Pipeline status Succeeded

View debug run consumption

All status

Monitor in Azure Metrics | Export to CSV

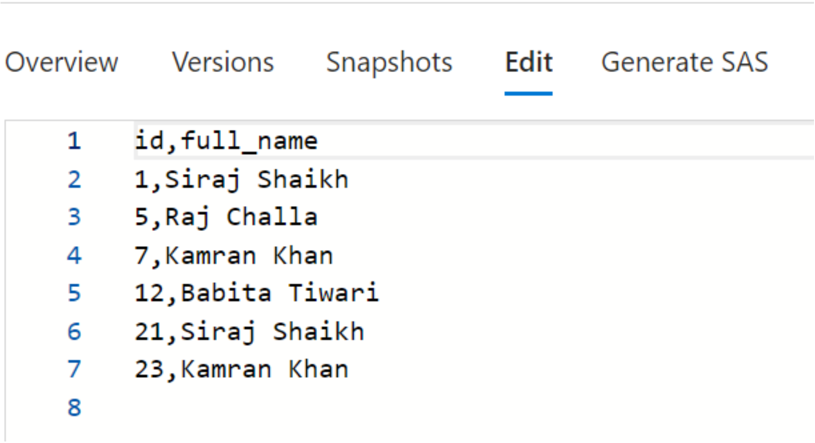
Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity
Data flow1	✓ Succeeded	Data flow	6/7/2024, 1:11:27 PM	1m 21s	AutoResolveIntegratior		049043

-Image 3

7. Verifying the Output

I then visited the output folder in the adf-training container and checked the output. The file was generated correctly (refer image 1).



```
id,full_name
1,Siraj Shaikh
5,Raj Challa
7,Kamran Khan
12,Babita Tiwari
21,Siraj Shaikh
23,Kamran Khan

```

-Image 1

Summary:

This document details the steps involved in creating a data flow in Azure Data Factory for unpivoting data.