

Data Engineering in the Cloud

Import Imperfectly Formatted CSV Files with ADF

Xuemao Zhang
East Stroudsburg University

January 18, 2025

Outline

- Prerequisites
- Lab

Prerequisites

- In this lecture, we create a data pipeline to import imperfectly formatted or unformatted CSV files.

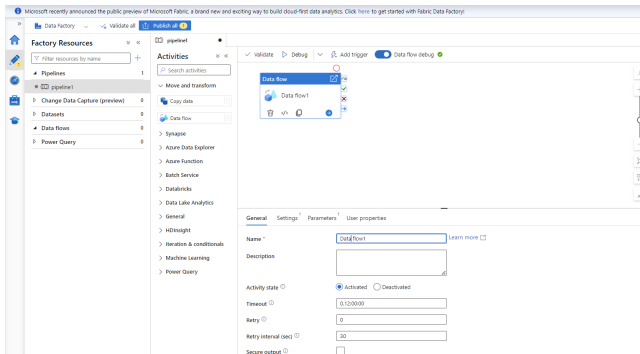
	A	B	C	D
	Date	TotalSales	TotalAmount	
2	12/8/2021	102	3060000	
3	15-09-2021	130	3900000	
4	25-10-2021	150	4500000	
5	2/11/2021	160	4800000	
6	3/11/2021	180	5400000	
7	8/11/2021	250	7500000	
8				
9				

Prerequisites

- Prerequisites (5 minutes):
 - ▶ Create a datalake (storage account with hierarchical namespace enabled)
 - ★ To lower the cost, you may choose Redundancy as LRS
 - ▶ Create two containers in the storage: `raw` and `output`
 - ★ Upload the data set `DP203.csv` to the container `raw`
 - ▶ Create an Azure Data Factory and launch Azure Data Factory Studio
- In the lab, we transform the data and then copy the transformed data to the container `output`

Lab

- 1 In the Azure Data Factory you just created, go to the “Author tab”
- 2 Click on the “Pipelines” to create a pipeline
 - And drag and drop the “Data Flow” activity onto the pipeline canvas



- 3 Click on + New to add a data flow

The screenshot displays the Azure Data Factory (ADF) interface. On the left, the 'Activities' pane is open, showing a search bar and a list of activities under 'Move and transform'. The 'Data flow' activity is highlighted. In the center, a 'Data flow' activity card is shown with a red circle highlighting the '+ New' button. Below the activity card, the 'Settings' tab is active, showing configuration options for the data flow. The 'Data flow' dropdown is set to 'Select...', and the 'Run on (Azure IR)' is set to 'AutoResolveIntegrationRuntime'. The 'Compute size' is set to 'Small'. The 'Logging level' is set to 'Verbose'. The 'Advanced' section is expanded, showing 'Sink properties' and 'Staging' options.

pipeline1

Activities

Search activities

Move and transform

Copy data

Data flow

Synapse

Azure Data Explorer

Azure Function

Batch Service

Databricks

Data Lake Analytics

General

HDInsight

Iteration & conditionals

Machine Learning

Power Query

Validate

Debug

Add trigger

Data flow debug

Data flow

Data flow1

General

Settings

Parameters

User properties

Data flow *

Select...

+ New

Run on (Azure IR) *

AutoResolveIntegrationRuntime

Compute size *

Small

Advanced

Logging level *

Verbose Basic None

Sink properties

Staging

Lab

- 4 In Source settings, click + New to add the data set Dp203.csv

The screenshot shows the Databricks Source settings for a dataset source named 'source1'. The interface includes a top bar with 'pipeline1' and 'dataflow1' tabs, and buttons for 'Validate', 'Data flow debug', and 'Debug Settings'. Below the top bar, there's a visual representation of the source as a blue arrow pointing right, labeled 'source1' with 'Columns: 0 total'. A dashed box below it contains the text 'Add Source' and a downward arrow. The main settings area has tabs for 'Source settings', 'Source options', 'Projection', 'Optimize', 'Inspect', and 'Data preview'. The 'Source settings' tab is active, showing fields for 'Output stream name' (set to 'source1'), 'Description' (set to 'Add source dataset'), 'Source type' (with 'Dataset' and 'Inline' options), 'Dataset' (a dropdown menu set to 'Select...' with a '+ New' button), and 'Options' (including 'Allow schema drift' which is checked, and 'Infer drifted column types' which is unchecked).

pipeline1 dataflow1

✓ Validate Data flow debug Debug Settings

source1
Columns:
0 total

Add Source

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * source1 [Learn more](#)

Description Add source dataset [Reset](#)

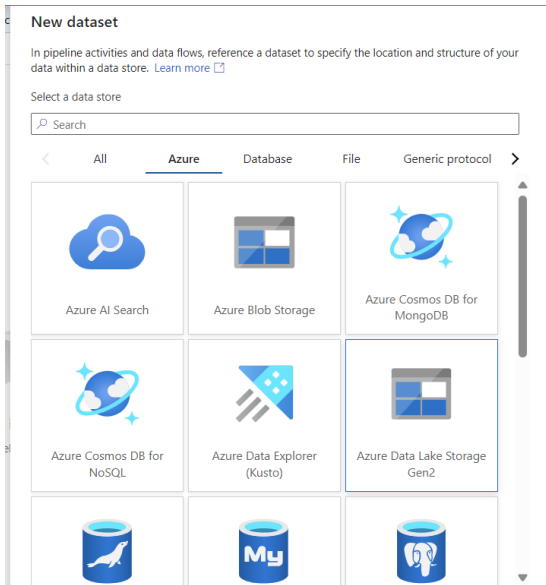
Source type * ☒ Dataset ☐ Inline

Dataset * Select... [+ New](#)

Options ☒ Allow schema drift ☐ Infer drifted column types

Lab

- 5 Select “Azure Gen-2 Storage account” and click on “Continue”











Lab

- 6 Select “DelimitedText” and click on “Continue”

Select format

Choose the format type of your data

 Avro	 Binary	 DelimitedText
 Excel	 JSON	 ORC
 Parquet	 XML	

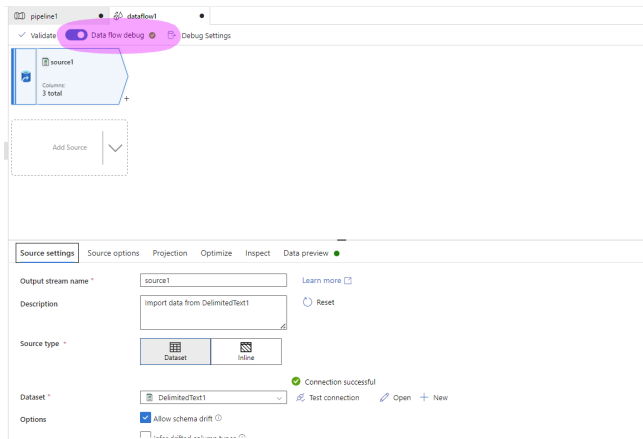
- A linked service needs to be created; Set properties as shown below and click on “OK”

The screenshot shows a 'Set properties' dialog box with the following fields and options:

- Name:** DelimitedText1
- Linked service *:** AzureDataLakeStorage1 (with a dropdown arrow and an edit icon)
- File path:** raw / Directory / DP203.csv (with folder and dropdown icons)
- First row as header:** ☒
- Import schema:** ☒ From connection/store, ☐ From sample file, ☐ None
- Buttons:** OK, Back, Cancel

Lab

- 8 Change the source name, select the source dataset,
 - ▶ with data flow debug option enabled (this may take some time)
 - ▶ then *Test connection*



- 9 Go to “projections” and you should be able to see the “schema”

The screenshot shows the Databricks workspace interface. At the top, there are tabs for 'DelimitedText1' and 'dataflow1'. Below these, there are buttons for 'Validate', 'Data flow debug' (which is active), and 'Debug Settings'. A sidebar on the left shows a tree view with 'source1' selected, indicating it has 3 columns. The main area has a tabbed interface with 'Projection' selected. Below the tabs, there are buttons for 'Define default format', 'Detect data type', 'Import projection', and 'Reset schema'. The 'Projection' tab displays a table with columns for 'Column name', 'Type', and 'Format'.

Column name	Type	Format
Date	abc string	Specify format
TotalSales	abc string	Specify format
TotalAmount	abc string	Specify format

- 10 Now click on “+” and click on “select”

The screenshot shows the Azure Data Factory 'dataflow1' editor. On the left, the 'Factory Resources' pane lists 'Data flows' with 'dataflow1' selected. The main canvas shows a data flow with a 'source1' source (3 total columns) and a 'sink1' sink. A '+' button is visible between them. A context menu is open over the '+' button, showing options like 'Join', 'Conditional Split', 'Exists', 'Union', 'Lookup', 'Schema modifier', 'Derived Column', 'Select', 'Aggregate', 'Surrogate Key', 'Pivot', 'Unpivot', 'Window', 'Rank', and 'External Call'. The 'Select' option is highlighted. On the right, the 'Data preview' section shows a table with columns for Date, TotalSales, and TotalAmount.

- 11 Configure the settings as shown below. Check/edit the mapping names to the columns

Publish all

DelimitedText1 • dataflow1 •

✓ Validate ☒ Data flow debug ☒ Debug Settings

source1
Input data from DelimitedText1

select1
Columns: 3 total

Add Source

Select settings Optimize Inspect Data preview ●

Output stream name * select1 [Learn more](#)

Description
Renaming source1 to select1 with columns 'Date, TotalSales, TotalAmount' [Reset](#)

Incoming stream * source1

Options
☒ Skip duplicate input columns
☒ Skip duplicate output columns

Input columns * ☐ Auto mapping [Reset](#) [Add mapping](#) [Delete](#) 3 mappings: All inputs mapped

	source1's column	Name as		
<input type="checkbox"/>	Date	→	Date	+ -
<input type="checkbox"/>	TotalSales	→	TotalSales	+ -
<input type="checkbox"/>	TotalAmount	→	TotalAmount	+ -

- 12 Now again click on “+” and select the “derived column” option

The screenshot shows the Apache Airflow web interface. A data pipeline is visible with a task named 'select1'. A context menu is open over the '+' icon next to the task, showing options for adding new tasks. The 'Derived Column' option is highlighted. The 'Data preview' tab is active, showing a table with 6 rows of data.

Date	TotalSales
NULL	102
NULL	130
NULL	150
NULL	160
NULL	180
NULL	250

Lab

- 13 Choose the column Date, open the expression builder, and select the expression you want to be formatted to.

The screenshot displays the Apache Airflow web interface for a task named 'DateTransform'. The top navigation bar includes 'Publish all' and 'dataflow1'. Below the navigation bar, there are tabs for 'Validate', 'Data flow debug', and 'Debug Settings'. The main canvas shows a workflow with three tasks: 'source1' (Import data from DelimitedText1), 'Reference: 1' (Columns: 3 total), and 'DateTransform' (Columns: 3 total). The 'DateTransform' task is selected, and its configuration is shown in the bottom panel. The 'Output stream name' is 'DateTransform', the 'Description' is 'Creating/updating the columns: 'Date', 'TotalSales', 'TotalAmount'', and the 'Incoming stream' is 'select1'. The 'Columns' section shows a table with two columns: 'Column' and 'Expression'. The 'Column' column has a dropdown menu with 'Date' selected. The 'Expression' column has a text input field with the placeholder 'Enter expression...' and a button 'Open expression builder'.

Derived column's settings

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

[+ Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

Column	Expression
<input checked="" type="checkbox"/> Date	<input type="text" value="Enter expression..."/> Open expression builder

Dataflow expression builder

derivedColumn1

Derived Columns

+ Create new

ANY Date

Column name *

Date

Expression

toDate(Date, 'dd-MM-yyyy')

+ - * / || && ! ^ == === <=> != >

Expression elements

All

Functions

Input schema

Parameters

Cached lookup

Data flow library functions

Locals

Expression values

Filter by keyword

+ Create new

ANY Date

ANY TotalSales

ANY TotalAmount

123 abs(123 numeric_value)

123 acos(123 numeric_value)

ANY add(ANY first_expression, ANY second_expression)

ANY addDate(ANY date@timestamp, ANY date.to.add)

Data preview

Refresh


Save and finish

Cancel

Clear contents

- 14 Click on “save” and “finish”

✓ Validate ☒ Data flow debug ☒ Debug Settings



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

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

+ Add [Clone](#) [Delete](#) [Open expression builder](#)

Columns * ○

<input checked="" type="checkbox"/> Column	Expression
<input checked="" type="checkbox"/> Date	<input type="text" value="toDate(Date, 'dd-MM-yyyy')"/> + -

- 15 Again, click on “+” and click on “Sink”

The screenshot displays the Databricks Dataflow Builder interface. At the top, there's a 'Publish all' button and a 'dataflow1' tab. Below this, a pipeline is shown with three components: 'source1' (Import data from DelimitedText1), 'select1' (Renaming source1 to select1 with columns 'Date', 'TotalSales', 'TotalAmount'), and 'derivedColumn1' (Columns: 3 total). A '+' button is visible next to 'derivedColumn1', which has been clicked to open a menu. The menu lists various actions: Window, Rank, External Call, Cast, Formatters (Flatten, Parse, Stringify), Row modifier (Filter), Sort, Alter Row, Assert, Flowlets (Flowlet), Destination, and Sink. The 'Sink' option is highlighted with a pink circle. Below the pipeline, the 'Derived column's settings' panel is open, showing fields for 'Output stream name' (derivedColumn1), 'Description' (Creating/Updating the columns 'Date', 'TotalSales', 'TotalAmount'), 'Incoming stream' (select1), and 'Columns' (Date).

Lab

- 16 Specify the output CSV folder
 - Make sure you choose the output container

The screenshot displays the Databricks Data Flow interface. At the top, the workflow is visualized as a sequence of steps: **source1** (Import data from DelimitedText1), **select1** (Renaming source1 to select1 with columns: Date, TotalSales, TotalAmount), **derivedColumn1** (Creating/updating the columns: Date, TotalSales, TotalAmount), and **sink1** (Columns: 3 total). Below the visualization, the **Sink** configuration panel is active, showing settings for the output stream **sink1**. The description is "Export data to DelimitedText2". The incoming stream is **derivedColumn1**. The sink type is set to **Dataset**. The output container is **DelimitedText2**, which is highlighted with a pink background. The connection status is "Connection successful". Other options include "Skip line count" and "Allow schema drift".

DelimitedText1 • dataflow1

✓ Validate Data flow debug Debug Settings

source1 Import data from DelimitedText1

select1 Renaming source1 to select1 with columns: Date, TotalSales, TotalAmount

derivedColumn1 Creating/updating the columns: Date, TotalSales, TotalAmount

sink1 Columns: 3 total

Add Source

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * sink1 [Learn more](#)

Description Export data to DelimitedText2 [Reset](#)

Incoming stream * derivedColumn1

Sink type * Dataset Inline Cache

Dataset * DelimitedText2 [Test connection](#) [Open](#) [New](#) [Connection successful](#)

Skip line count

Options ☒ Allow schema drift ☐ Validate schema

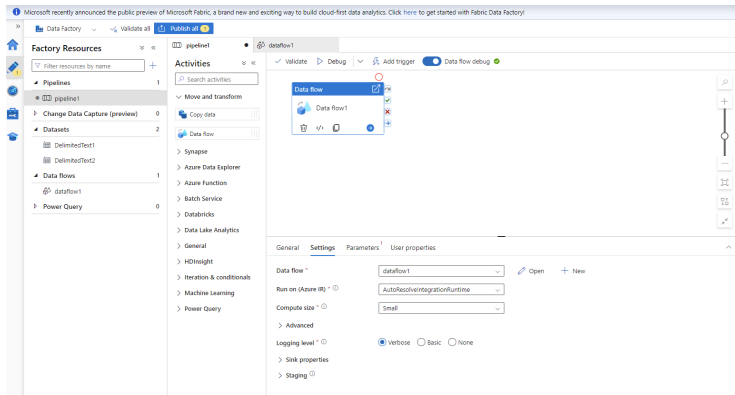
- 17 Select file name option as default

The screenshot displays a data pipeline configuration interface. At the top, a workflow is shown with four stages: 'source1' (Import data from DelimitedText1), 'select1' (Renaming source1 to select1 with columns: 'Date', 'TotalSales', 'TotalAmount'), 'derivedColumn1' (Creating/updating the columns: 'Date', 'TotalSales', 'TotalAmount'), and 'sink1' (Columns: 3 total). Below the workflow, a dashed box contains an 'Add Source' button. The 'Settings' tab is selected, showing various configuration options:

- Clear the folder:** ☐
- File name option *:** Default (dropdown)
- Quote All ①:** ☐
- Headers ①:** Enter expression... (text input) ANY
- Umask ①:**
 - Owner: ☐ R ☐ W ☐ X
 - Group: ☐ R ☒ W ☐ X
 - Others: ☐ R ☒ W ☐ X
 - Octal: 022 (text input)
- Pre/post commands ①:**
 - File pre command: mkdir, mv, cp, rm (text input) + [trash icon]
 - File post command: mkdir, mv, cp, rm (text input) + [trash icon]

Lab

- 18 Now click on “Publish all” button
- 19 Now go to the “Pipelines” tab and you should be able to see the complete pipeline



- 20 Click on the “debug” and you will see a pipeline generated below:

The screenshot displays the Azure Data Factory (ADF) console. At the top, there's a 'Publish all' button. Below it, the pipeline 'dataflow1' is selected. The 'Activities' pane on the left shows a search bar and a list of activities under 'Move and transform', including 'Copy data' and 'Data flow'. The main canvas shows a single activity 'Data flow1' of type 'Data flow'. The bottom section shows the 'Pipeline run ID' as 'cb3cac4-96b3-425c-a1a0-fdab949b3d4d' and the 'Pipeline status' as 'In progress'. Below this, a table lists the activities in the pipeline run.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	Use
Data flow1	In progress	Data flow	7/24/2024, 7:00:31 PM	6s		

Lab

- 21 Click on “add trigger” and “trigger now” button to trigger the pipeline. This will take some time to trigger.
- You can go to the output container to preview the output data.

The screenshot shows the Microsoft Azure portal interface. On the left, the 'output' container is selected under 'Containers'. The 'Overview' tab is active, showing the authentication method as 'Access key' and the location as 'output'. A search bar is present with the text 'Search blobs by prefix (case-...)'. Below the search bar, there is a list of blobs. The blob 'part-00000-02e3406b-3aa4-4beb-914e-20db27de9aed-c000.csv' is selected, and a context menu is open over it, showing options like 'View/edit', 'Download', 'Properties', 'Generate SAS', 'Manage ACL', 'Change tier', 'Acquire lease', 'Break lease', 'Delete', and 'Rename'.

part-00000-02e3406b-3aa4-4beb-914e-20db27de9aed-c000.csv

Overview Versions Edit Generate SAS

1	Date	TotalSales	TotalAmount
2	2021-08-12	102	3060000
3	2021-09-15	150	3900000
4	2021-10-25	150	4500000
5	2021-11-02	100	4800000
6	2021-11-03	100	5400000
7	2021-11-08	250	7500000
8			

License



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).