



ADF Template – Synapse Link - Dataverse to SQL

Microsoft





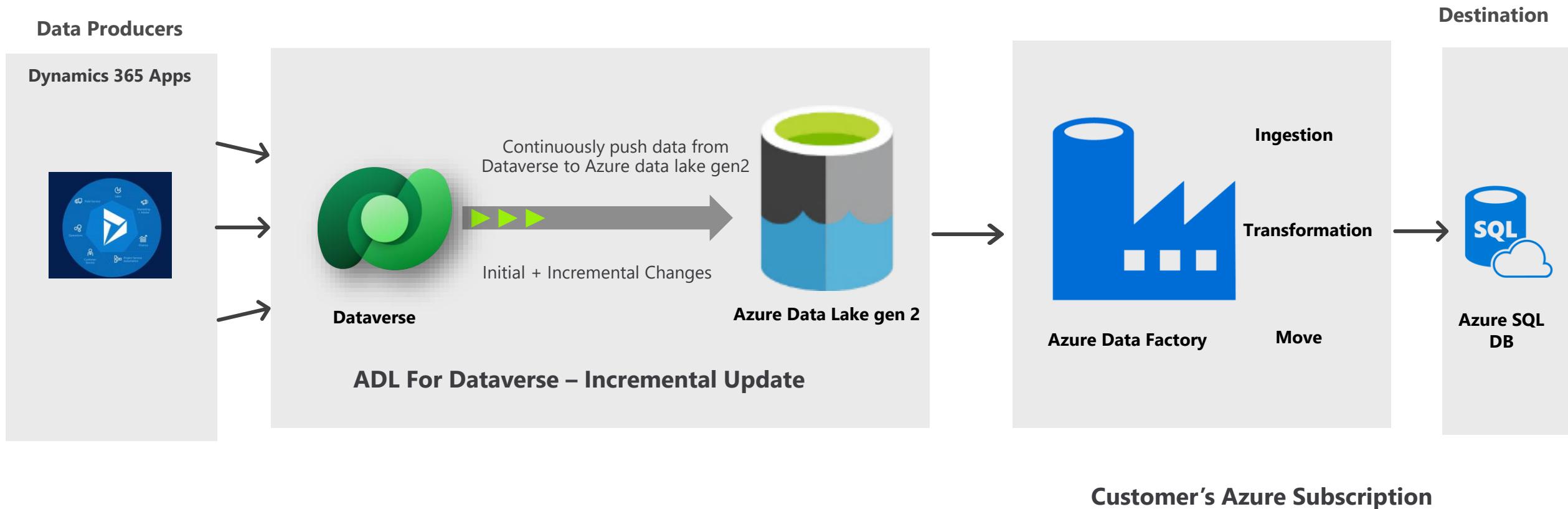
Note:

Content and dates presented in this document are non-binding and subject to change without any prior notice.

Session Objectives

- Solution Overview
- ADF Pipeline Template
- FAQ and Best Practices
- Known Issues
- Appendix

Solution Overview



Solution Overview

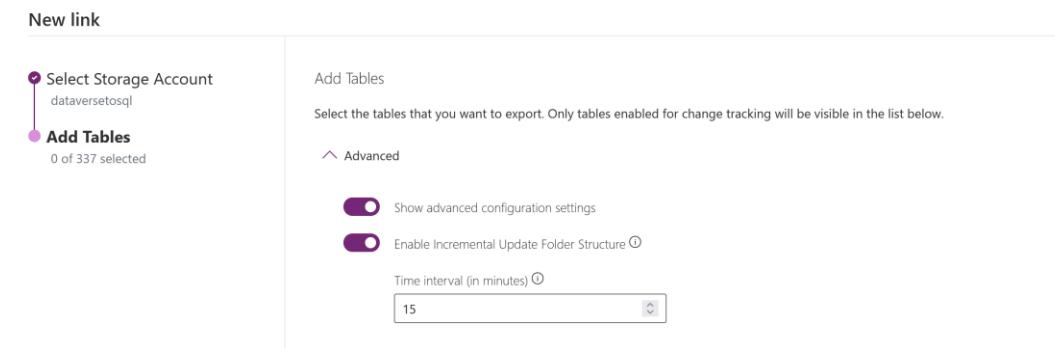
- Cost effective
 - Trigger based
 - Number of triggers based on time period selected
 - Optimized network egress: only latest changes are moved.
- Low maintenance
 - Automated table schema sync
 - DeleteLog support
- Automatic infrastructure management
 - Combine pipeline intent w/ resource allocation & mgmt
- Operational Excellence:
 - Telemetry systems, usage optimization, etc.

Solution Overview

- Reuse existing SQL assets (DES)
 - Table schema in similar pattern like DES tables.
 - Minimal disruption experience.

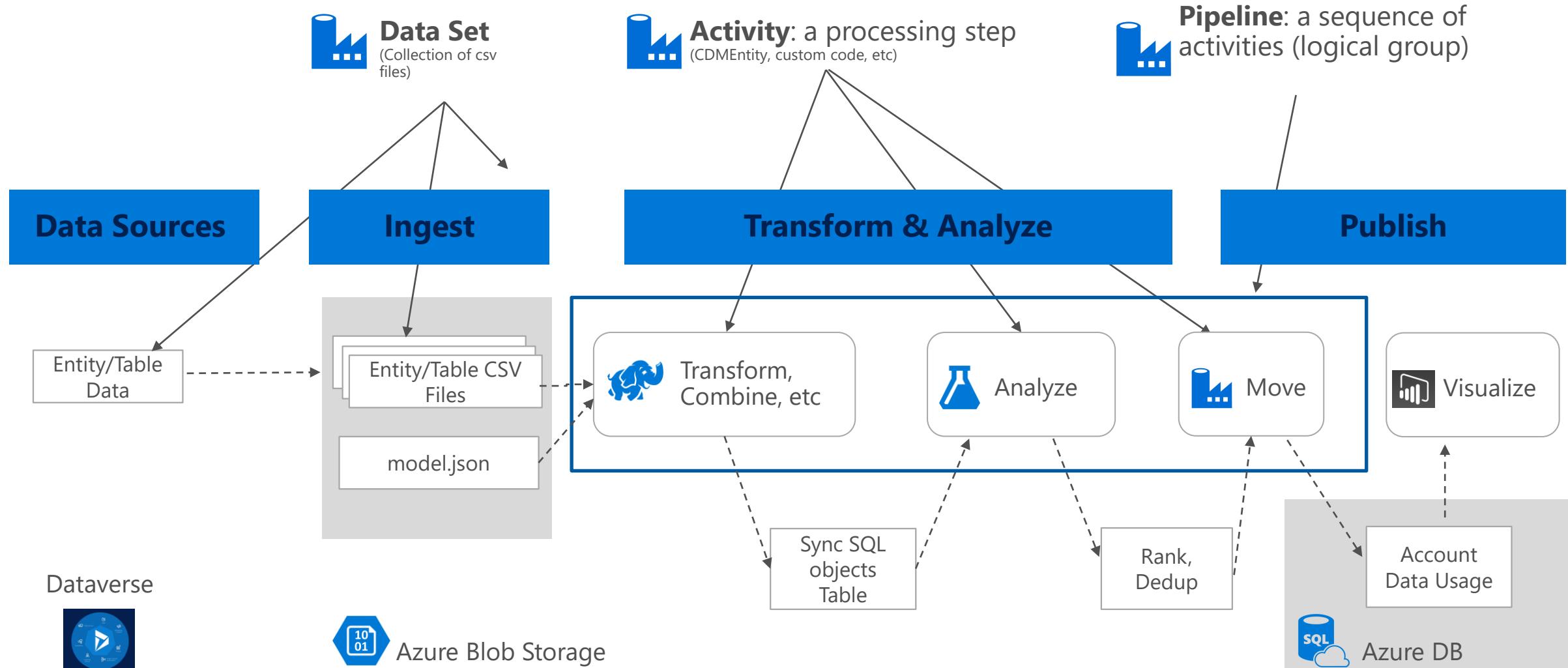
Solution Overview – Incremental Updates

- Drop stale and stagnant data to save data processing costs and increase efficiency in Azure Synapse Analytics.
- Track data changes during a user-specified time period.
- Timestamp and table folders are created only when there is a data update during the user-specified time interval.



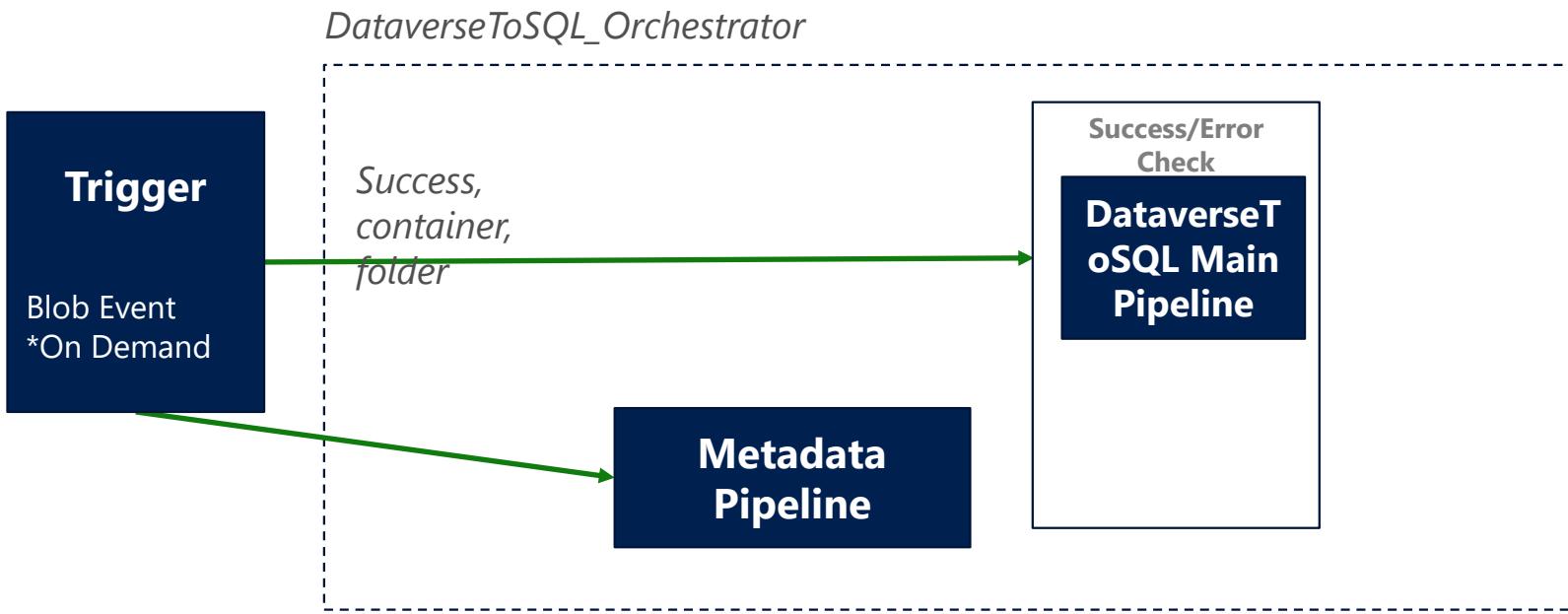
<input type="checkbox"/>	📁 2022-07-21T22.41.33Z
<input type="checkbox"/>	📁 2022-07-21T22.56.33Z
<input type="checkbox"/>	📁 2022-07-21T23.11.33Z
<input type="checkbox"/>	📁 2022-07-22T00.26.32Z
<input type="checkbox"/>	📁 2022-07-22T00.41.33Z
<input type="checkbox"/>	📁 2022-07-25T23.11.34Z
<input type="checkbox"/>	📁 2022-07-25T23.26.32Z
<input type="checkbox"/>	📁 2022-07-26T00.41.33Z
<input type="checkbox"/>	📁 2022-07-28T02.41.32Z
<input type="checkbox"/>	📁 2022-07-28T02.56.33Z

Solution Overview - Data Factory Concepts



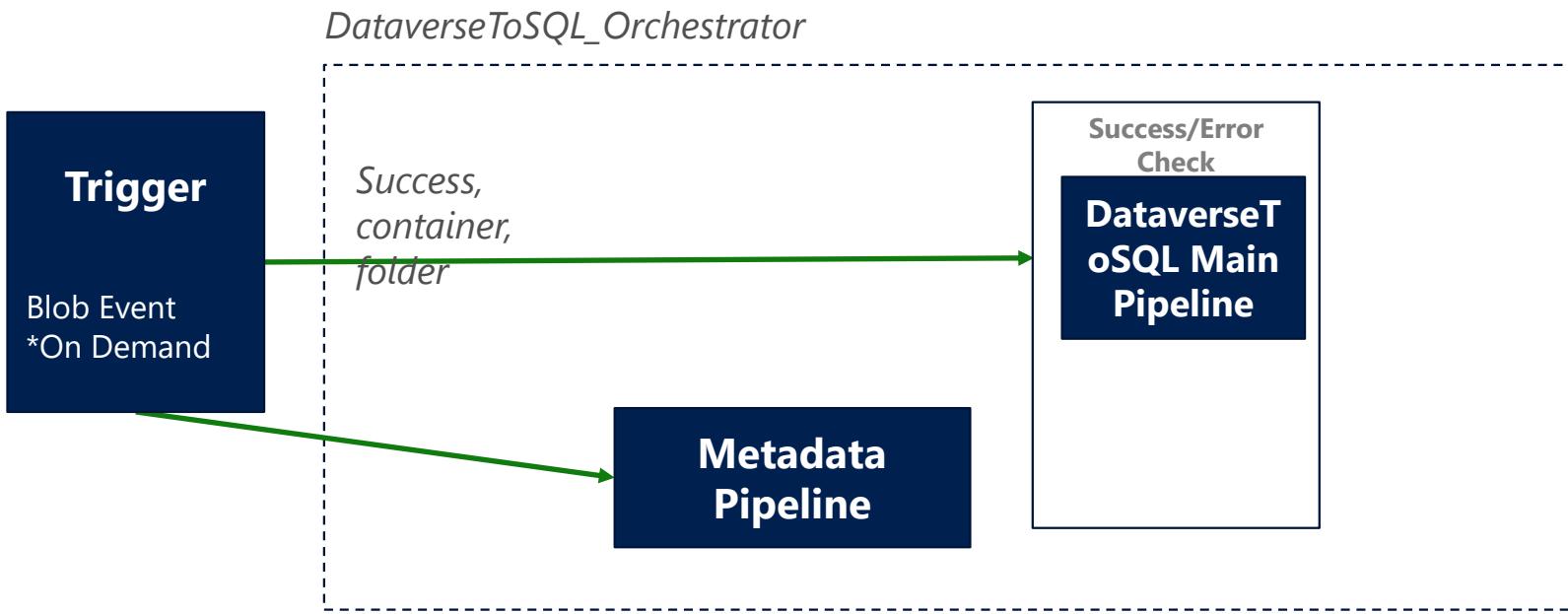
Azure Data Factory – Orchestrator Pipeline

Validates previous pipeline runs and executes the main pipeline-
DataverseToSQL and OptionsetMetadata



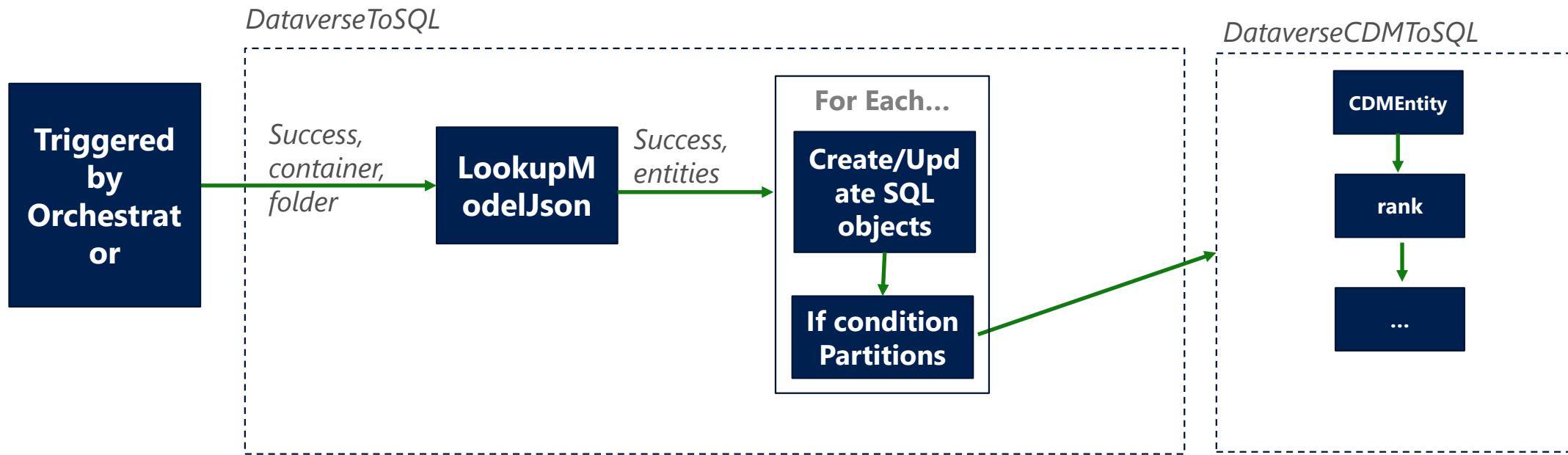
Azure Data Factory – Orchestrator Pipeline

Validates previous pipeline runs and executes the main pipeline-
DataverseToSQL and OptionsetMetadata

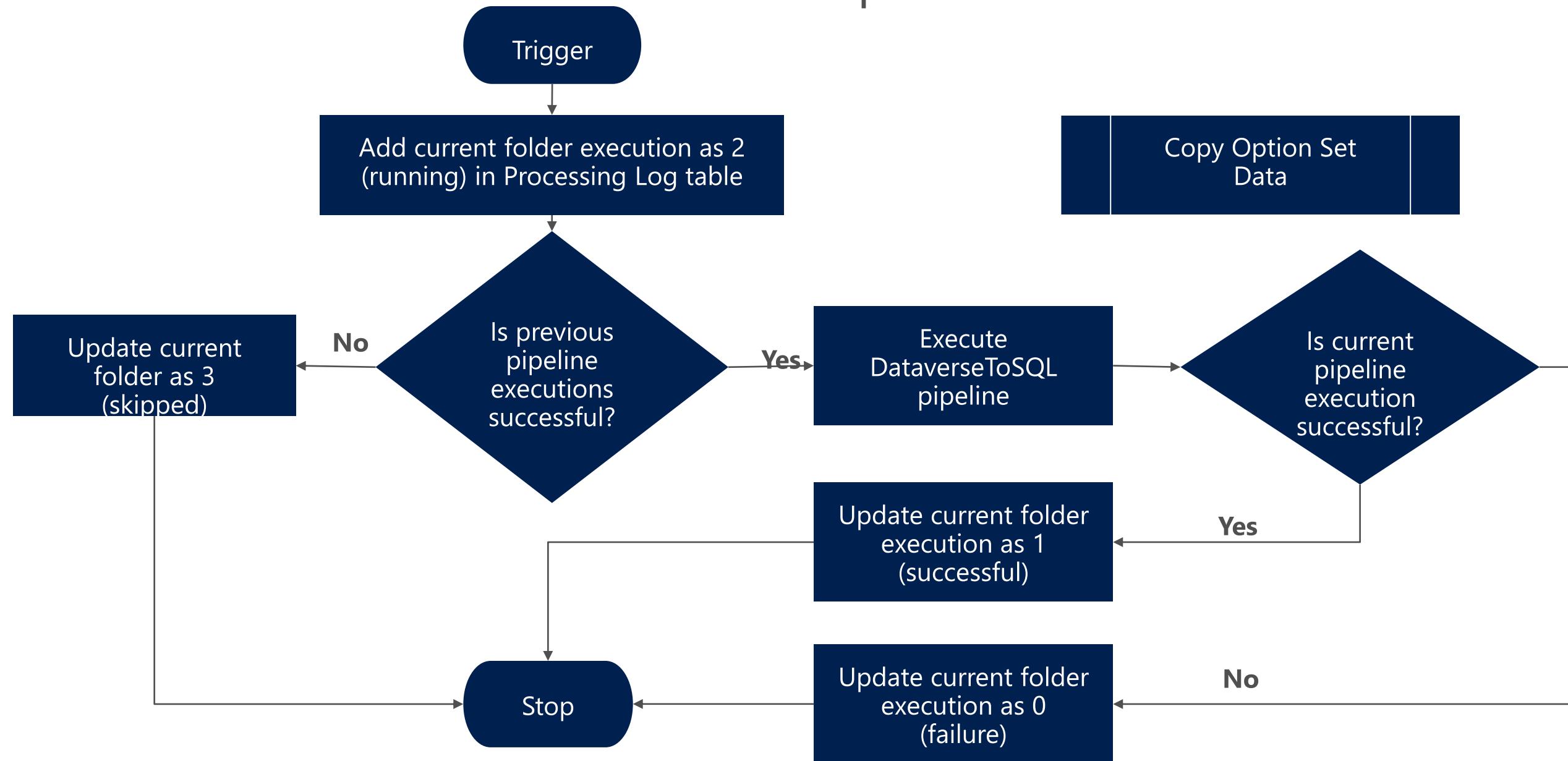


Azure Data Factory – DataverseToSQL Control Flow

Coordinate pipeline activities into finite execution steps with looping, conditionals and chaining. Separated data transformations in individual data flow



Orchestrator Pipeline Flow



Discoverability

FAQs

Q: Where is the pipeline available for download?

Answer: Please download the ADF template from ADF Gallery.

Q: Does the pipeline work with ADF & Synapse Workspace?

Answer: Yes. However, documentation has steps for ADF Studio. Deployment steps are similar for Synapse workspace.

Q: Is there public documentation for deploying the ADF template?

Answer: Yes, Please refer to <https://learn.microsoft.com/en-us/power-apps/maker/data-platform/azure-synapse-link-pipelines?tabs=synapse-analytics>

ADF Template

FAQs

Q: How many templates are there?

Answer: There are three (3) templates.

1. DataverseToSQL_Orchestrator
2. DataverseToSQL
3. CopyOptionsetMetadata

FAQs

Q: What is the purpose of these pipelines?

Answer: Orchestrator checks the success of previous pipeline runs and executes the other two pipelines.

Orchestrator pipeline handles below scenarios:

- Error/Failure
- Concurrency

DataverseToSQL is the main pipeline to process the data

CopyOptionsetMetadata is for option set metadata.

FAQs

Q: What are the statuses DataverseToSQL_Operator pipeline uses?

Answer: Status in Processing Log table

0=Failure

1=Success

2=Running (this is temporary status and it eventually become 0,1 or 3)

3=Skipped

FAQs

Q: Why is there separate pipeline to copy option set metadata?

Answer: OptionSet metadata files are not part of model.json hence they will not be copied by the DataverseToSQL template.

Q: Does the trigger needs to be added to all pipelines?

Answer: No. Only the orchestrator pipeline.

Q: Does ADF template need parametrization of the individual folder that would update every 15 min to run?

Answer: No. Once trigger is added for the storage account, Synapse Link will trigger the ADF pipeline.

FAQs

Q: Can the DataverseToSQL pipeline be triggered manually?

Answer: Yes. In manual trigger, provide actual container name and folder name as parameter value.

The screenshot shows the 'Pipeline run' interface in the Azure Data Factory portal. On the left, the 'Activities' pane lists various data integration services like Move & transform, Azure Data Explorer, Azure Function, etc. The main area displays a 'Lookup' activity. At the top right, a message says 'Trigger pipeline now using last published configuration.' Below it, the 'Parameters' section shows two entries:

Name	Type	Value
container	string	dataverse-devorg2-unq3cd6 ...
folder	string	2022-09-28T14.11.33Z

FAQs

Q: Can pipeline execution continue after the next pipeline is triggered?

Answer: No. Pipeline is designed to process single folder at a time. Concurrent pipeline execution for same entity/table will lead to duplicates and errors. To avoid concurrency issue, ensure that pipeline is optimized so that current execution completes before next trigger.

Orchestrator pipeline will skip execution of DataverseToSQL pipeline for all subsequent folders until all the previous folders are successfully processed (status =1). This is necessary to avoid data integrity issues.

FAQs

Q: Why is concurrency an issue for the pipeline execution?

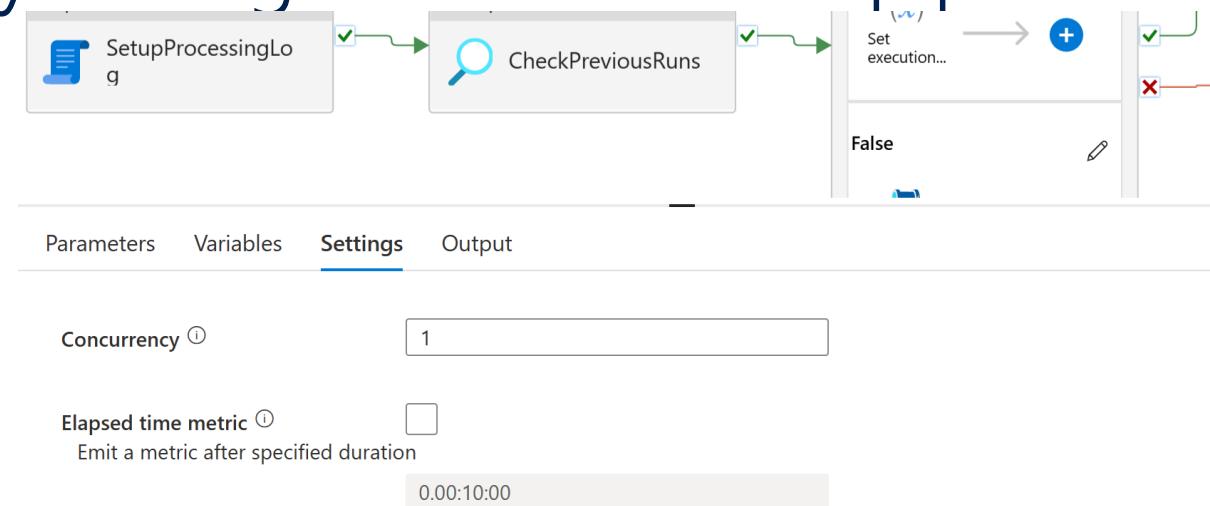
Answer: To avoid stale/old data overriding the latest data, there is check for max(versionnumber) for a given table/entity.

If two pipeline execution is running concurrently and table/entity data from recent folder gets processed before the previous one, then older folder's data won't get processed since that would have lower version number. This will lead to data loss.

FAQs

Q: With delta sync sometimes, it takes more than 15 mins and all the pipelines after that are getting skipped. It is leading to a lot of manual effort. Can we avoid folders getting skipped in this scenario?

Answer: Yes. To avoid folders getting skipped update the concurrency setting of orchestrator pipeline to 1.



FAQs

Q: With delta sync sometimes, it takes more than 15 mins and all the pipelines after that are getting skipped. Can we avoid folders getting skipped in this scenario (contd)?

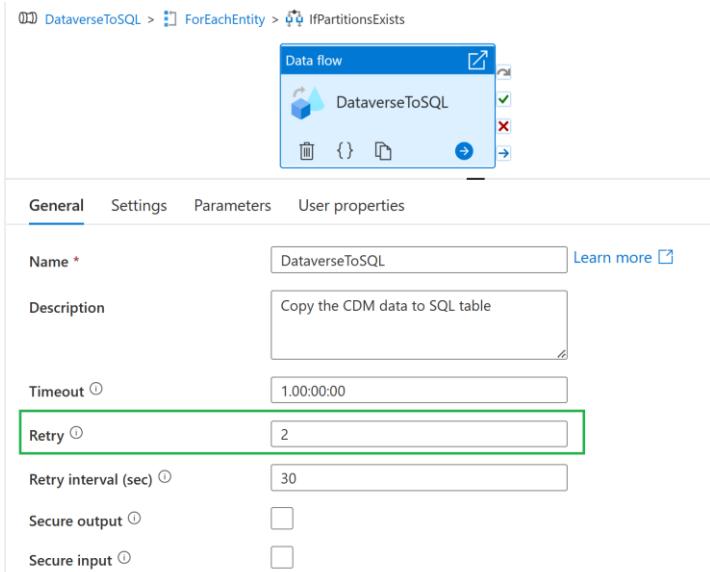
Answer: With concurrency = 1. Subsequent Orchestrator pipeline runs will get queued. It is important to ensure pipeline execution completes within reasonable time window else the subsequent pipeline runs will get queued.

Showing 1 - 5 items	Last run ended 2 minutes ago				
<input type="checkbox"/> Pipeline name ↑↓	Run start ↑↓	Run end ↑↓	Duration	Triggered by	Status ↑↓
<input type="checkbox"/> DataverseToSQL_Orchestrator	1/24/2023, 8:57:01 PM	--	00:00:00		 Queued
<input type="checkbox"/> DataverseToSQL	1/24/2023, 8:55:36 PM	--	00:03:14		 In progress
<input type="checkbox"/> CopyOptionsetMetadata	1/24/2023, 8:55:23 PM	1/24/2023, 8:55:45 PM	00:00:21		 Succeeded
<input type="checkbox"/> DataverseToSQL_Orchestrator	1/24/2023, 8:55:22 PM	--	00:03:28		 In progress

FAQs

Q: There are always going to be transient failures – either in SQL connections or apparently in the IR instances. Each time one of these failures occur, it just kills the entire sync process. What is the best way to handle these kind of transient errors?

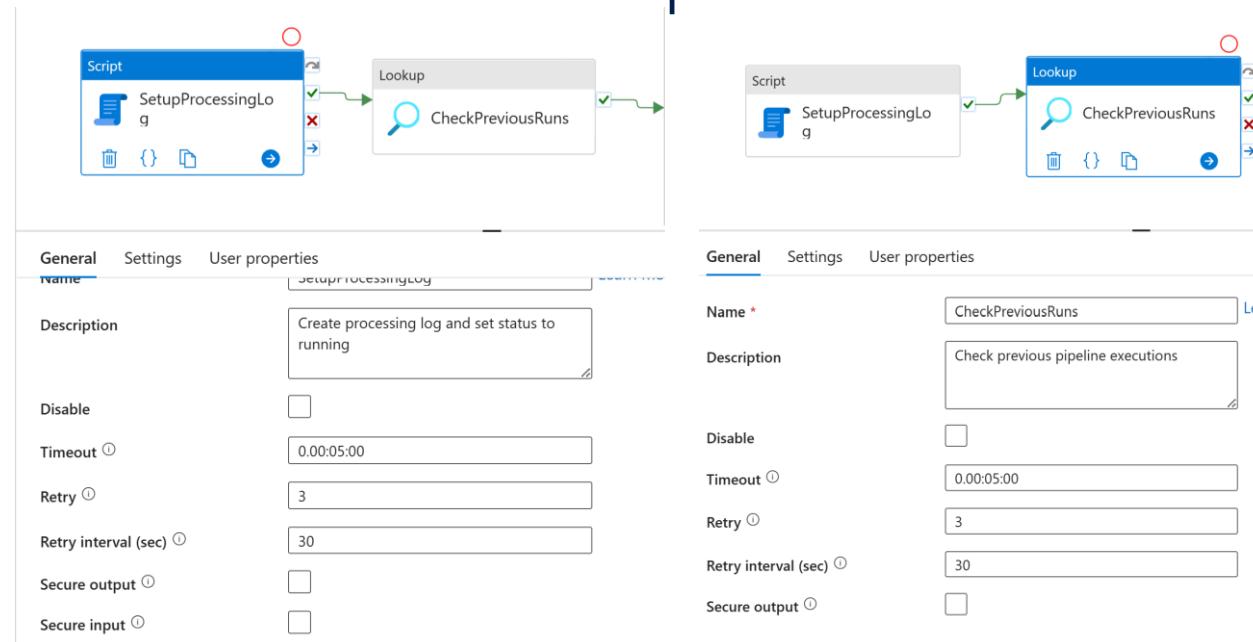
Answer: Consider appropriate retry count & time out for the data flow.



FAQs

Q: There are always going to be transient failures – either in SQL connections or apparently in the IR instances. Each time one of these failures occur, it just kills the entire sync process. What is the best way to handle these kind of transient errors (contd)?

Answer: Consider appropriate retry count & time out for all pertinent activities. Sample below



* timeout & retry value should be carefully chosen to ensure balance between robustness & performance.

FAQs

Q: How can long running/stuck pipeline be dealt with?

Answer: Please ensure appropriate timeout value for the data flow.

FAQs

Q: Why does Orchestrator pipeline shows success even though it skipped processing a folder?

Answer: This is fair feedback. Consider enhancing the orchestrator pipeline by adding Fail activity which executes if main pipeline has skipped execution. Pipeline variable that can be used in this scenario is "executionskipped"

FAQs

Q: A folder was missed to get processed; can it be manually reprocessed by the ADF template?

Answer: If future folders are not processed (status=3), then process the folders in chronological order.

Please see troubleshooting section for more details.

ADF template checks for latest version number so older version of rows cannot be processed by the ADF template.

FAQs

Q: Can a folder (missed) be processed outside of ADF logic?

Answer: Yes. However, it would require extensive manual steps. Evaluate the below steps:

- Copy the DataverseToSQL pipeline and use a prefix that is not used (for example “temptest_”).
- Execute the DataverseToSQL pipeline for the missed folder. This would result in data for the folder to be copied to “temptest_” tables.
- Process the deletes, inserts and updates using SQL queries.
- For update ensure that only the rows in “temptest_” tables with higher versionnumber than destination/main table is considered

FAQs

Q: We have observed our activitypointer table slows down execution for all other entities, is it possible to process in a dedicated pipeline execution without configuring another Synapse Link?

Answer: Yes. However, there would be maintenance overhead. Evaluate the below steps:

- Copy the DataverseToSQL & DataverseToSQL_Orchestrator pipeline and name them as DataverseToSQL_ap & DataverseToSQL_Orchestrator_ap
- Use a prefix in orchestrator pipeline (for example "ap_").
- Update DataverseToSQL_Ochestraor_ap to execute DataverseToSQL_ap and remove CopyOptionsetMetadata

FAQs

Q: We have observed our activitypointer table slows down execution for all other entities, is it possible to process in a dedicated pipeline execution without configuring another Synapse Link?

Answer: (contd).

- Update DataverseToSQL pipeline to exclude activitypointer by using filter.
- Update DataverseToSQL_ap to process only activitypointer using filter
- Add same trigger to DataverseToSQL_Orchestrator_ap as DataverseToSQL_Orchestrator. *Reconfirm the filters of the trigger.

FAQs

Q: Does ADF pipeline support custom DB schema & table prefix?

Answer: Yes, please update the variables in the pipeline. Don't forget the trailing underscore for the prefix "_".

Q: Does ADF pipeline removes the processed folders & files?

Answer: No. Please plan to remove the processed folders after confirmation of data in destination DB.

FAQs

Q: For the Azure SQL sink in the Data Flow, does Use TempDB needs to be checked/unchecked?

Answer: When Use TempDB is checked, ADF would use Global Temporary table instead of physical table.

Consensus is that physical table will have better performance than temp table. However, it is possible that performance difference is very negligible hence it best to evaluate performance of the pipeline with Use TempDB checked/unchecked and make an informed decision.

FAQs

Q: For the Azure SQL sink in the Data Flow, does partition option needs to updated?

Answer: Please check one of AzureSQLDB diagnostics for a data flow run to determine if partitions are optimal. If default setting is creating higher than required partitions, consider reducing them.

<https://learn.microsoft.com/en-us/azure/data-factory/connector-azure-sql-database?tabs=data-factory#mapping-data-flow-properties>

FAQs

Q: There is single DeleteLog table for all entities/tables, will it cause bottleneck?

Answer: Deletes are expected to be less and only operation is Insert hence it shouldn't be a bottleneck.

Please consider having the Single Partition for AzureSQLDBDeleteLog sink.

FAQs

Q: Why is there additional duplicate check in data flow?

Answer: Additional (redundant) check was introduced to ensure (double) that duplicate rows doesn't cause key violation error. However, this has increased the number of activity in the data flow hence it is planned to be removed in future release. There will be no loss of functionality.

FAQs

Q: We noticed that DataverseToSQL pipeline loops through all the entities even where there is no data for them. Can this be avoided?

Answer: Main data flow is not executed if there are no partitions in the model.json. All the entities are still looped to ensure that table schema is synced even if there is no data.

If syncing table schema for entities that don't have data is not a requirement, then please add filter to remove entities with no partitions.

*see the sample pipeline update in the Appendix section for a design approach

Option Set Metadata

FAQs

Q: Data Export Service only brings the values of globaloptionset. However, in Azure Synapse link it brings in the value from all the tables wherein a globaloptionset is being used. Does the values need to deduplicated before syncing it to Azure SQL?

Answer: No. GlobalOptionSetMetadata in Synapse Link has additional entity name column. Entity name should be used in the joins. Values shouldn't de-duplicated.

Destination Database

FAQs

Q: Does the destination Azure SQL DB needs to be in same GEO?

Answer: No. However, it is recommended to have all the Azure resources co-located.

Q: Can destination table be optimized?

Answer: Yes, destination table is created based on information available in model.json file and might not be optimized for specific workload. Please add required indexes for query optimization.

Adding index to versionnumber column is highly recommended

FAQs

Q: Can DES table be used?

Answer: No. DES schema is slightly different than the schema available in model.json.

Pipeline execution will have errors if DES tables are used because alter statements will fail (constraints error etc.)

Incremental Update

FAQs

Q: Incremental Update feature works in append-only mode, will we get multiple rows?

Answer: No. Pipeline has logic to get only the latest rows and only latest rows will be pushed to destination Azure SQL tables.

Q: Will only incremental data be available?

Answer: When Incremental update is configured, there will be a full sync of data from Dataverse. There is no filter option here, initial sync will bring all data for the selected entities to ADLS.

FAQs

Q: What is the steps to remove and readd entity?

Answer:

- Remove entity for Synapse Link.
- Delete the table in destination Azure SQL.
- Re-add entity to the Synapse Link.

FAQs

Q: Our count in make.powerapps.com->Synapse Link & Azure SQL DB doesn't match!

Answer: This is expected & by-design. Incremental Update feature works in append-only mode with all transactions being synced to the storage account. Count displayed in make.powerapps.com can be higher than number of records in the Dataverse environment & destination Azure SQL DB.

FAQs

Q: Is Synapse Workspace and/or serverless DB supported for Incremental Update?

Answer: No. This is by-design.

This feature can't be enabled with the option: **Connect to your Azure Synapse workspace**. For customers who require Azure Synapse analytics access, follow this

Please refer to documentation

<https://learn.microsoft.com/en-us/power-apps/maker/data-platform/azure-synapse-incremental-updates>

FAQs

Q: Can two or more Synapse link profile created in same environment?

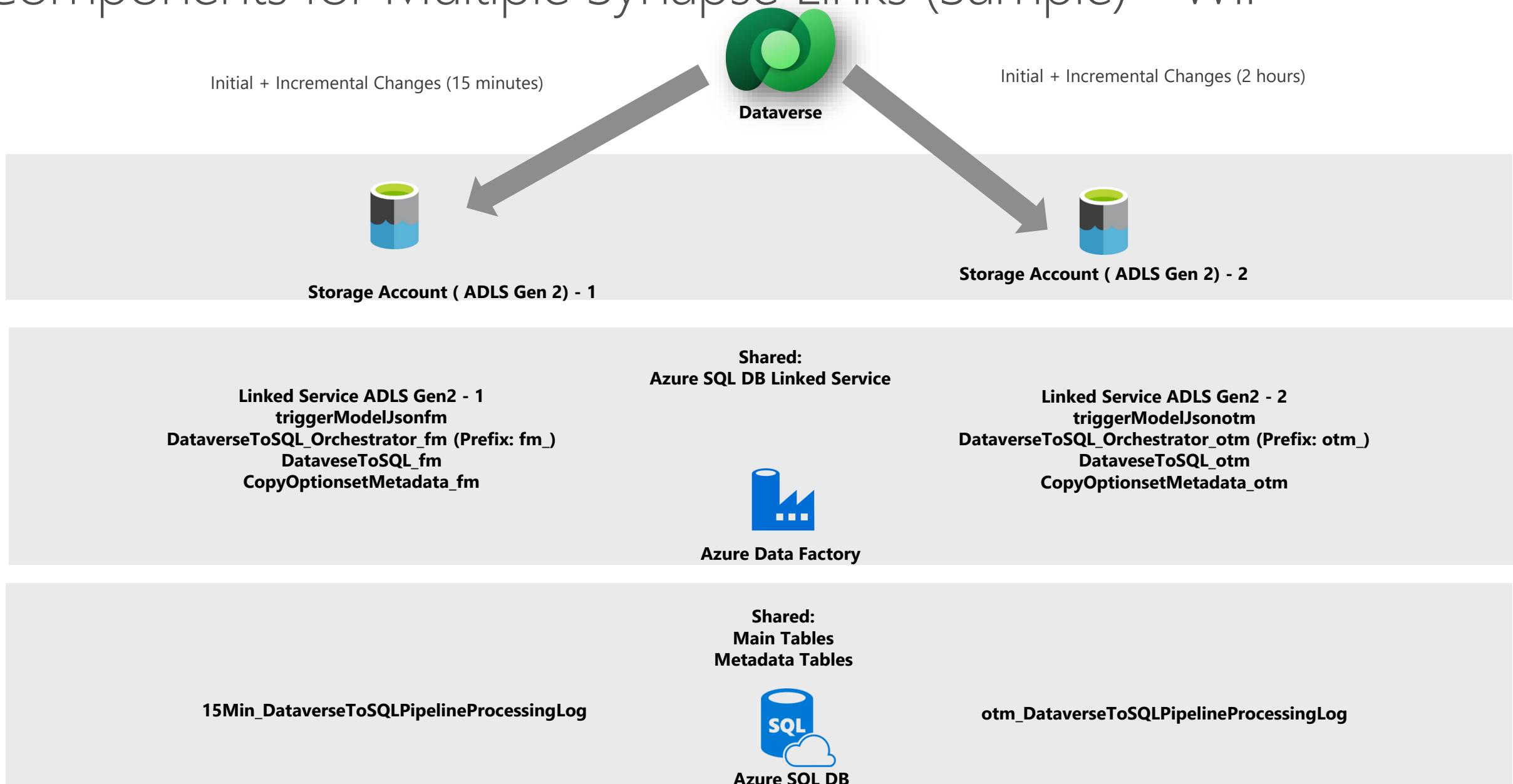
Answer: Yes.

Q: Can ADF template process two different Synapse link and point to same Azure SQL DB?

Answer: Yes. Please consider below points:

1. Each Synapse link should be configured for different time interval.
2. Consider having different Orchestrator pipeline and DataverseToSQLPipelineProcessingLog table. Copy the orchestrator pipeline and use different prefix in each.

Components for Multiple Synapse Links (Sample) - WIP



FAQs

Q: Can incremental folders removed/cleaned from the ADLS?

Answer: Yes. Processed folder can be removed without any impact to the sync process. Ensure that active folder is not impacted.

Q: What are the options to remove processed folders?

Answer: There several options to archive/delete folders from ADLS.

FAQs

Q: Why SinkCreatedOn & SinkModifiedOn columns in csv files have same values?

Answer: This is by-design. The data csv file in the incremental in the incremental folders will have same SinkCreatedOn & SinkModifiedOn value.

Azure Data Lake Storage

FAQs

Q: Can pipeline trigger be stopped/paused & then restarted?

Answer: Yes. However, it is not recommended to stop/pause the trigger in production.

Please ensure storage filters are in place after the trigger is restarted. Additionally, folders completed while trigger was stopped/paused won't be auto-processed.

Table Schema (Dataverse Metadata Sync)

FAQs

Q: How is destination table created?

Answer: Destination table is created based on information available in model.json file and data type mapping in script component

Q: How is column mapping handled?

Answer: Column mapping based on data type specified in model.json file and SQL script in script component.

FAQs

Q: How can customer prevent metadata sync if they want it to remain constant (and managed manually)?

Answer: We do not recommend this way, as it will incur more risk and maintainability going forward. The reason - There can be intermittent failure in the pipeline as metadata structure has changed in Dataverse and the data transformation needs to manually incorporate. This will also require additional debugging and can be time consuming. Please note that updates to existing fields & new fields can be introduced as part of weekly updates/release wave.

FAQs

Q: We understand the risk of stopping the auto metadata sync, how can the metadata sync be stopped?

Answer: Below statements needs to be commented out in script component of DataverseToSQL pipeline:

```
EXECUTE sp_executesql @AlterTableAddColumnDDL;
```

```
EXECUTE sp_executesql @AlterTableAlterColumnDDL;
```

FAQs

Q: How is data type change handled? For example, in Dataverse, a string field was dropped & recreated as int.

Answer: This is not a supported scenario. Synapse link will stop working if attribute/columns data type is changed.

FAQs

Q: Each model.json only contains the tables and columns defined in that specific Synapse Link definition - are any child or dependent tables pulled in automatically or only the ones they specifically define?

Answer: Only the tables selected in the Synapse Link definition will be synced. Dependent tables won't be synced automatically.

FAQs

Q: DES table had createdon as datetime and Synapse link has it as datetimeoffset(7), should we modify the ADF template to accommodate the change.

Answer: No. datetimeoffset has higher precision than datetime. There would be data loss.

Please consider casting/conversion in SQL views/store procedure to accommodate changes to the data type.

Initial Data Load

FAQs

Q: Does ADF Template handle initial sync as well?

Answer: Yes, if trigger has been added along with Incremental Update – Synapse Link setup.

Q: If trigger was not added, is there any recommended way to migrate existing data from Synapse Link (Data Lake) to SQL?

Answer: Yes. Chronologically & sequentially process the existing folders. Once all the existing folders are processed, add the trigger so that new folders will be auto-processed.

FAQs

Q: Initial data sync generates very high volume and pipeline takes more than 15 minutes to execute, what can be done for initial load?

Answer: Few pointers for initial load

Consider having highest DTU tier for Azure SQL for initial load. DTU tier can be lowered after the initial load

Consider having custom IR with appropriate (minimum) TTL. IR can be switched after initial load completes.

Add the large entities in Synapse link one at a time and before the smaller entities. If larger entities are added before smaller entities, the ForEach loop execution will faster since model.json won't lots of entities in it.

FAQs

Q: Initial data sync generates very high volume and pipeline takes more than 15 minutes to execute, what can be done for initial load (contd)?

Answer: Few pointers for initial load (contd).

Plan for manual monitoring for the duration of initial load. Please ensure concurrency for the orchestrator pipeline is set to 1 so that if any pipeline run takes additional time, subsequent folders shouldn't get skipped.

Supportability

FAQs

Q: What is the support model for the ADF template?

Answer: This ADF template is a code sample. You are more than encouraged to use this template as a guidance to test out the functionality of retrieving data from ADLS Gen 2 storage to Azure SQL DB using the pipeline provided.

However, we are always open to getting your feedback!

Please provide any suggestions or questions about this template in our forums ADF Stack Overflow

(<https://stackoverflow.com/questions/tagged/azure-data-factory>) or Microsoft Q&A Portal

(<https://docs.microsoft.com/en-us/answers/topics/azure-data-factory.html>).

FAQs

Q: How would updates to the ADF template be handled?

Answer: Once ADF Pipeline is downloaded from ADF Gallery in customer's Azure subscription, updates published will not impact existing pipelines.

Implementation's team can download the latest version based on their schedule.

FAQs

Q: We encountered unexpected error with pipeline execution. What should we do?

Answer: Please create an Azure support request.

<https://learn.microsoft.com/en-us/azure/azure-portal/supportability/how-to-create-azure-support-request>

FAQs

Q: We encountered unexpected error with Synapse Link. What should we do?

Answer: Please create a support request in Power Platform admin center.
<https://learn.microsoft.com/en-us/power-platform/admin/get-help-support>

Tell us about the issue

What product were you using when the issue occurred? *

Dynamics 365 Customer Engagement

Tell us what you need help with *

Type a brief description, question or error message. Text will be used to recommend solutions. Do not include personal data or confidential/proprietary information.

Error for a specific entity/table in Azure Synapse Link for Dataverse

187/256 characters remaining

Problem type *

Export to Data Lake

Environment *

<https://devorg0.crm.dynamics.com/> X

Azure Synapse Link for Dataverse >

Tables Details Discover hub

Table ↑	Name	Sync status	Last synchronized on	Count	Append only	Partition
Account	[REDACTED]	Active	02/23/2023 11:42:13 AM	124780	Yes	Year
Activity Purpose	[REDACTED]	Active	02/21/2023 8:05:43 AM	30	Yes	Year
Address	[REDACTED]	Active	02/23/2023 11:42:13 AM	6011811	Yes	Year
Case	[REDACTED]	Active	02/23/2023 11:42:13 AM	4124371	Yes	Year
Case Resolution	[REDACTED]	Active	02/23/2023 11:43:27 AM	4124371	Yes	Year
Client Program	[REDACTED]	Active	02/23/2023 11:41:51 AM	281474	Yes	Year
Connection	connection	Error		849926	Yes	Year
Connection Role	[REDACTED]	Active	02/21/2023 12:06:41 PM	347	Yes	Year
Contact	[REDACTED]	Active	02/23/2023 11:42:03 AM	1838109	Yes	Year

Best Practices

Best Practices

- Confirm the Azure region of Dataverse environment before provisioning Azure Storage account.
- Ensure Dataverse, Azure Storage, ADF & Azure SQL are all in the same Azure region for better performance (lower latency) & minimize costs (reduce data movement cost) <https://azure.microsoft.com/en-us/pricing/details/bandwidth/>

Best Practices

- Have right sized destination Azure SQL database. Higher DTU will give better performance.
- Initial sync retrieves historical data which will have higher data volume.
- Confirm the acceptable time period for data sync with business upfront.
- Plan for multiple Synapse link if there are multiple groups of entities that need to be synced in different time intervals.
- Plan for ADF execution costs before production deployment.

Best Practices

- Validate Create, Update, Delete operations for a single entity/table.
- Validate metadata changes from Dataverse to destination SQL DB.
- Start with a low volume entity to ensure:
 - Trigger and trigger filters are appropriately set
 - Orchestrator pipeline triggers for appropriate folder
 - All the linked services can connect

Best Practices

- Review the pipeline runs to identify opportunity for optimization.
- For higher parallel processing, consider increasing the batch size of ForEach loop in DataverseToSQL pipeline.
- Update the Azure SQL sink's partition option (under optimize) to make sure too many partitions are not been created.
- Consider reducing cluster start up by creating Integration Run which has appropriate (minimum) Time to live.

Best Practices

- Use new destination database for the ADF pipeline (instead of DES database). This will help:
 - Run DES & ADF pipeline simultaneously
 - Avoid resource constraints in the destination DB.
 - Avoid table schema conflicts since DES table schema won't match 100% with schema provided in model.json

Best Practices

- Evaluate all Azure SQL best practices.
- <https://learn.microsoft.com/en-us/azure/azure-sql/database/automatic-tuning-overview?view=azuresql>

Best Practices

- Plan to monitor ADF pipelines by setting relevant alerts
- <https://docs.microsoft.com/en-us/azure/data-factory/monitor-metrics-alerts>
- <https://docs.microsoft.com/en-us/azure/data-factory/monitor-visually#alerts>
- <https://learn.microsoft.com/en-us/azure/data-factory/concepts-annotations-user-properties>
- Recommended alerts:
- Any pipeline failure
- If execution time exceeds “x” minutes. (“x” will depend on the time period selected for incremental folder)
- <https://learn.microsoft.com/en-us/azure/service-health/alerts-activity-log-service-notifications-portal>

Best Practices

- Plan and manage cost
- <https://learn.microsoft.com/en-us/azure/data-factory/plan-manage-costs>
- <https://learn.microsoft.com/en-us/azure/data-factory/plan-manage-costs#monitor-costs-at-pipeline-level-with-cost-analysis>
- <https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/cost-mgt-alerts-monitor-usage-spending>

Performance Optimization

Performance Optimization Check List

- Destination Azure SQL DTU.
- Indexes for the Azure SQL tables.
- Batch size of ForEach loop in DataverseToSQL pipeline.
- Dataflow - Azure SQL sink's partition option (under optimize).
- ADF Integration Runtime - Time to live.

Cost Optimization

Cost Optimization Check List

- Incremental Update – Time Window
 - Higher time window will reduce the pipeline runs and reduce cost.
 - Consider multiple Synapse Links if entities/tables need to sync in different time windows is required.
- Batch size of ForEach loop in DataverseToSQL pipeline.
 - Choose lowest batch size that is optimal. Batch size controls how many spark cluster will be started simultaneously.
- ADF Integration Runtime - Time to live
 - Keep TTL to minimum. Having TTL for 5 minutes will be more cost effective than having it at 15+ minutes.

Cost Optimization Check List

- ADF Integration Runtime – Core Count
 - Increasing core count has shown minimal performance increase hence it is recommended to go with minimum number of cores (4x4).
- Minimizing Pipeline Activities
 - Evaluate if table schema sync is needed if there is no data for certain entities/tables in given incremental folder. Please refer to sample for filtering out entities/tables without partition(s).

Settings

Settings Check List

- Region/Geo:
- Dataverse, Azure Storage, ADF & Azure SQL should be in same geo/region.
- Azure SQL:
- Appropriate DTU/vCores
- ADF Integration Runtime:
- Compute Type: Memory optimized
- Core Count: Minimum (4x4).
- DataverseToSQL_Orchestrator pipeline
- Concurrency: 1
- Timeout of pertinent activities: Appropriate value
- Retry of pertinent activities: More than 1 (appropriate value)

Settings Check List

- CreateTable - Script:
- Timeout: Appropriate value
- Advanced – Script block execution timeout (minutes): Appropriate value

Settings Check List

- DataverseToSQL pipeline
- ForEach:
- Batch count: Appropriate value
- DataverseToSQL Data flow Activity (inside ForEach):
- Timeout: Appropriate value
- Retry: More than 1 (appropriate value)
- Other Activities:
- Timeout of pertinent activities: Appropriate value
- Retry of pertinent activities: More than 1 (appropriate value)
- DataverseCDMToSQL Data Flow
- AzureSQLDBDeleteLog sink -> Number of partitions: 1 or 2
- AzureSQLDB sink -> Number of partitions: Appropriate value
- AzureSQLLog sink -> Number of partitions: Appropriate value

Troubleshooting

Gen2 operation failed for: 'filesystem' does not match Path split(item()?pipelineParameters?FolderPath, '/')[1] is outside bounds

Root Cause 1: Incremental update feature has not been configured.

Sample error:

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, there's a sidebar with navigation links for Analytics pools, Activities, and Integration. The main area is titled 'Trigger runs' and shows a table of recent runs. The table includes columns for Trigger name, Trigger time, Status, Pipelines, Trigger file name, Trigger file size, Message, and Properties. One row is highlighted in grey, indicating a failure. The 'Status' column shows a red 'Failed' icon. The 'Message' column contains the error message: "'The template language expression 'split(item()?pipelineParameters?FolderPath,'/')[1]' cannot be evaluated because array index '1' is outside bounds (0, 0) of array.'" The 'Properties' column shows a gear icon, which when clicked, reveals the detailed error message.

Trigger name	Trigger time	Status	Pipelines	Trigger file name	Trigger file size	Message	Properties
triggerModelJson	Nov 28, 2022, 9:52:15 pm	Failed	0	/blobServices/default/containers/datasets	124731 B	Error "The template language expression 'split(item()?pipelineParameters?FolderPath,'/')[1]' cannot be evaluated because array index '1' is outside bounds (0, 0) of array."	
triggerModelJson	Nov 28, 2022, 9:52:07 pm	Succeeded	1	/blobServices/default/containers/datasets			
triggerModelJson	Nov 28, 2022, 9:37:21 pm	Failed	0	/blobServices/default/containers/datasets			
triggerModelJson	Nov 28, 2022, 9:37:13 pm	Succeeded	1	/blobServices/default/containers/datasets			
triggerModelJson	Nov 28, 2022, 9:22:17 pm	Failed	0	/blobServices/default/containers/datasets			
triggerModelJson	Nov 28, 2022, 9:22:08 pm	Succeeded	1	/blobServices/default/containers/datasets			

`split(item()?pipelineParameters?FolderPath, '/')[1]` is outside bounds

Root Cause 2: Filters for the trigger is missing.

Resolution: Please validate the filter for the ADF Pipeline trigger. Filters are documented towards the end of the documentation <https://learn.microsoft.com/en-us/power-apps/maker/data-platform/azure-synapse-link-pipelines?tabs=synapse-analytics>

The trigger might be getting executed for updates to model.json under parent container.

Sample error:

Trigger name ↑↓	Trigger type	Trigger time ↑↓	Status ↑↓	Pipelines	Run	Message
trigger1	Storage events tri...	Nov 15, 2022, 8:46:06	✗ Failed	0	Original	View
trigger1	Storage events tri...	Nov 15, 2022, 8:45:31	✓ Su			
trigger1	Storage events tri...	Nov 15, 2022, 12:31:0	✗ Fa			
trigger1	Storage events tri...	Nov 15, 2022, 12:30:3	✓ Su			

Error

"The template language expression '`split(item()?pipelineParameters?FolderPath, '/')[1]`' cannot be evaluated because array index '1' is outside bounds (0, 0) of array."

Pipeline failure

Symptom: Pipeline encountered unexpected error and failed for the given folder.

DataverseToSQLPipelineProcessingLog has entry of 0 (failure) for the failed folder. Subsequent folders are marked as 3 (skipped).

Resolution:

1. Identify & resolve the root cause of pipeline failure.
2. Manually run the DataverseToSQL pipeline for the failed folder.
3. After successful execution of manual run, update the corresponding row in DataverseToSQLPipelineProcessingLog to 1 (success)

Pipeline failure (contd.)

4. Chronologically, sequentially & manually process the skipped (3) folders and manually update the DataverseToSQLPipelineProcessingLog Status column of corresponding rows to 1 (success).

Once the failed folder & all the skipped folder are marked as successful, DataverseToSQL_Orchestrator pipeline will automatically process the next folder in next trigger.

Pipeline failure (contd.)

Q: There are several skipped folders and manually executing them to updating DataverseToSQLPipelineProcessingLog is time consuming and error prone. Is there any better approach?

Answer: Yes. Please consider having new pipeline which:

1. Retrieves the skipped folder chronologically
2. Executes the DataverseToSQL pipeline sequentially
3. Updates Status column of DataverseToSQLPipelineProcessingLog rows to 1 (success).

*Sample pipeline in Appendix section.

Pipeline failure (contd.)

Q: If we delete folder since the failure and all line in DataverseToSQLPipelineProcessingLog after the same date, do you think the process will cover the gap/skipped folders?

Answer: No. Orchestrator pipeline works with the trigger based on model.json hence updating the DataverseToSQLPipelineProcessingLog table won't help with execution of skipped folders.

Pipeline failure (contd.)

Status in DataverseToSQLPipelineProcessingLog table

	Container	Folder	PipelineRunId	ProcessingStarted	ProcessingEnded	Status
10	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-10-30T18.56.33Z	d7c3cff9-d8fc-4cab-96ae-19a98b48e9ff	2022-10-30 22:41:51.3066667	2022-10-30 22:46:30.1500000	1
11	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-10-30T22.41.33Z	21ca7b3c-95b4-4e8b-ac4b-4b558d9dac91	2022-10-31 17:56:40.6633333	2022-10-31 18:01:47.5333333	1
12	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-10-31T17.56.32Z	91b9fd34-cdac-49ae-95f0-abe9378d22bc	2022-11-01 09:12:36.4000000	2022-11-01 09:16:28.1900000	1
13	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-01T09.11.32Z	7d72a38c-0101-47df-affb-5235b5890b96	2022-11-01 09:26:44.6233333	2022-11-01 09:30:47.0066667	1
14	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-01T09.26.33Z	a4d3c5f3-5f17-4edd-bfd3-f463479b2040	2022-11-02 07:27:26.6166667	2022-11-02 07:31:34.2200000	1
15	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-02T07.26.32Z	aaf9337c-0df6-4378-be39-08497410e95b	2022-11-02 23:57:26.3533333	2022-11-03 00:02:20.6200000	0
16	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-02T23.56.32Z	1cca9f81-7e69-475b-99a6-0f49515f116b	2022-11-04 20:56:42.1733333	2022-11-04 20:56:53.7833333	3
17	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-04T20.56.32Z	f0ceea43-c3e1-4271-909b-26d09107b71c	2022-11-04 21:26:42.6600000	2022-11-04 21:26:55.0500000	3
18	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-04T21.26.32Z	38397297-ef17-4197-96d6-834b48cffb98	2022-11-05 02:26:43.6266667	2022-11-05 02:26:54.2200000	3
19	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-05T02.26.32Z	1caedda3-293f-4be6-92b6-66f6213654f0	2022-11-05 20:27:27.1000000	2022-11-05 20:27:36.1966667	3
20	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-05T20.26.33Z	c232c390-9e4d-4628-870afea789e70d0d	2022-11-06 00:41:57.6200000	2022-11-06 00:42:05.6400000	3
21	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-06T00.41.33Z	24db930a-06bc-472b-97a4-65541d30edcd	2022-11-07 15:42:22.5800000	2022-11-07 15:42:32.6833333	3
22	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-07T15.41.33Z	69e99951-9a39-40da-b980-75379c165da9	2022-11-07 15:56:41.8866667	2022-11-07 15:57:01.0233333	3
23	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-07T15.56.32Z	73bf17be-63be-4c19-806a-185e283a8561	2022-11-07 18:11:42.8933333	2022-11-07 18:11:52.2333333	3
24	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-07T18.11.32Z	5c2eeef3a-0279-4fe6-837b-332849f36557	2022-11-11 21:12:23.5466667	2022-11-11 21:12:32.3800000	3
25	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-11T21.11.32Z	05b4f8c1-970f-4148-83ad-bf04c81e19f6	2022-11-12 00:56:50.1300000	2022-11-12 00:56:59.8800000	3
26	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-12T00.56.33Z	fc7844a6-7e13-4911-a310-6d8ea2a18baa	2022-11-12 02:11:47.2133333	2022-11-12 02:11:55.9633333	3
27	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-12T02.11.32Z	db14020c-66ee-4f37-8dd3-ba879587461d	2022-11-13 19:42:28.4666667	2022-11-13 19:42:46.7166667	3
28	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-13T19.41.32Z	53643f77-6c30-45de-a7c4-2cd4cb7b59f6	2022-11-14 22:12:23.9133333	2022-11-14 22:12:43.8666667	3
29	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-14T22.11.33Z	bd9aca03-ec14-416b-9608-f11fe0ad4353	2022-11-17 15:12:49.4966667	2022-11-17 15:12:58.7933333	3
30	dataverse-devorg2-unq3cd674d243f4ec11811d0022480cd	2022-11-17T15.11.33Z	ce04b6bc-9137-4b3c-9ef9-db95dc8838ff	2022-11-17 15:26:54.2200000	2022-11-17 15:27:03.4066667	3

No data is syncing, and orchestrator is not executing
DataverseToSQL pipeline

Pipeline getting skipped without any failure

Root Cause: ADF pipeline execution time overlapped, and
Orchestrator didn't execute main pipeline.

Sample log table entries:

167	dataverse-f	2022-11-17T08.09.32Z	27a0e124-7e38-4ec8-9707-a25aae699c7	2022-11-17 08:24:44.8000000	2022-11-17 08:38:51.3200000	1
168	dataverse-f	2022-11-17T08.24.32Z	69f02905-9e02-4d5e-9308-4e1f4de0a2bb	2022-11-17 08:39:46.2466667	2022-11-17 09:03:59.9300000	1
169	dataverse-f	2022-11-17T08.39.32Z	a00f0ffa-01a0-4ac3-8063-3362b5178dc2	2022-11-17 08:54:44.3800000	2022-11-17 08:55:00.9366667	3
170	dataverse-f	2022-11-17T08.54.32Z	40174c6c-d360-4ec8-b585-6f3c2d6ec803	2022-11-17 09:09:46.2133333	2022-11-17 09:10:04.3366667	3

Start of new pipeline (row 169) is before end of
previous pipeline (row 168). This caused concurrency
issue

No data is syncing, and orchestrator is not executing
DataverseToSQL pipeline

Pipeline getting skipped without any failure

Resolution:

1. Identify the root cause of why pipeline execution time exceeded.
2. Remediate the root cause to avoid similar issue for the future pipeline runs.
3. Process the skipped folders chronologically & sequentially.
4. Once the skipped folders are successfully processed, orchestrator will start executing the main DataverseToSQL pipeline.
5. Ensure that Concurrency setting of Orchestrator pipeline is 1 to avoid this scenario in future.

*see the sample pipeline in the Appendix section for a design approach

Re-syncing existing data (complete) for all the tables from Data Lake

"I want to rehydrate/replay all records from data lake."

Pre-requisites:

1. All the incremental folders are available in storage account.
2. Any table/entity was not removed & readded from make.powerapps.com.
3. DataverseToSQLPipelineProcessingLog table has rows for all the folders with ProcessingStarted column having date & time in chronologically order.
4. ADF trigger is turned on and Orchestrator pipeline is continually adding rows in DataverseToSQLPipelineProcessingLog table.

Re-syncing existing data (complete) for all the tables from Data Lake (contd.)

"I want to rehydrate/replay all records from data lake."

Resolution:

1. Update the first row in DataverseToSQLPipelineProcessingLog status from 1 to 3.
2. Wait for trigger to execute and ensure that processing of new folder starts getting skipped. Don't stop the trigger, it is critical to ensure that Orchestrator pipeline continue to make entry in DataverseToSQLPipelineProcessingLog table.
3. Mark all the status of non "3" rows as "3".
4. Truncate all the transactional tables like account, contact etc. Do not truncate the DataverseToSQLPipelineProcessingLog table.
5. Execute the Skipped pipeline*.

*see the sample pipeline in the Appendix section for a design approach

Re-syncing existing data (complete) for all the tables from Data Lake (contd.)

"I want to rehydrate/replay all records from data lake."

Resolution:

6. Skipped pipeline might be needed to run more than once to catch up with new folders.

Validating counts

"I still only have x records in the SQL database and Dataverse has y records."

Resolution:

1. Please understand that data is being copied in batch mode hence there is possibility of latest data yet to be synced.
2. Ensure custom logging is implemented in the ADF template.
3. Confirm there is no failure in DataverseToSQLPipelineProcessingLog table.
4. Run SQL query to get the count of rows. Sample using a log table:

```
-- Get distinct records excluding the deleted records
SELECT COUNT(DISTINCT CL.RecordId)
FROM [logdb].[dbo].[contactlog] CL WITH(NOLOCK)
WHERE NOT EXISTS
(SELECT 1 FROM [maindb].[dbo].[DeleteLog] DL WITH(NOLOCK) WHERE DL.RecordId =
CL.RecordId
AND EntityName = 'contact')
```

Validating counts

"I want to check how many records were inserted/updated/deleted for a given entity and folder."

Resolution:

1. Ensure custom logging is implemented in the ADF template.
2. Run SQL query to get the count of records group by folder. Sample using a log table:

```
-- Gets record count group by folder  
-- Add delete clause to exclude deleted records  
SELECT COUNT([RecordId]) AS RecordsCount, Folder  
FROM [dbo].[contactlog] CL WITH(NOLOCK)  
--WHERE IsDelete IS NULL  
GROUP BY CL.Folder
```

Delay in record sync

"I want to check when a given record was synced."

Resolution:

1. Ensure custom logging is implemented in the ADF template.
2. Run SQL query to get the log details for a given record. Sample:

```
-- Gets log details for a given record
-- JOIN DataverseToSQLPipelineProcessingLog for pipeline start & end details
-- Pipeline run id can also be used to search in log analytics
SELECT CL.RecordId, CL.LogDate, CL.SinkModifiedOn, *
FROM [dbo].[contactlog] CL WITH(NOLOCK)
WHERE CL.RecordId = '5a850971-9ab7-ed11-b596-00224802a7c1'
ORDER BY CL.LogDate DESC
```

Pipeline failure due to script failure in DataverseToSQL

Root Cause: Alter statement for very large table can time out.

Resolution:

1. High & appropriate time out for the script component
2. High & appropriate execution time out for the script component

The screenshot displays two components from an Azure Data Factory pipeline:

- Script Component:** This component contains a single action: "CreateTable". It is configured with a **Timeout** of **0:2:00:00**. Other settings include **Retry** (2), **Retry interval (sec)** (30), and **Secure output** and **Secure input** checkboxes.
- If Condition Component:** This component has a condition "IfPartitionsExists". The "True" path contains a "DataverseToSQL" action. The "False" path is currently empty.

Script Component Settings:

Setting	Value
Name *	CreateTable
Description	Create or alter table to destination SQL DB
Timeout	0:2:00:00
Retry	2
Retry interval (sec)	30
Secure output	<input type="checkbox"/>
Secure input	<input type="checkbox"/>

If Condition Component Settings:

Condition	Action
IfPartitionsExists	True: DataverseToSQL
IfPartitionsExists	False: (empty)

DB collation error

Cannot resolve the collation conflict between "Japanese_CI_AS" and "SQL_Latin1_General_CI_AS" in the equal to operation. Cannot resolve the collation conflict between "Japanese_CI_AS" and "SQL_Latin1_General_CI_AS" in the equal to operation.

Root Cause: Script activity doesn't have default collation set.

- add Columns

```
Declare @AlterTableAddColumnDDL nvarchar(max) = (SELECT  
'alter table ' + @entity + ' ADD ' + STRING_AGG(CONVERT(NVARCHAR(max), + '[' + name + '] ' + datatypeSQL), ',')  
from @metadata m  
left join INFORMATION_SCHEMA.COLUMNS c on  
c.TABLE_NAME = @entity and TABLE_SCHEMA = 'dbo'  
and c.COLUMN_NAME = m.Name COLLATE DATABASE_DEFAULT  
where c.COLUMN_NAME is null  
);
```

```
execute sp_executesql @AlterTableAddColumnDDL;
```

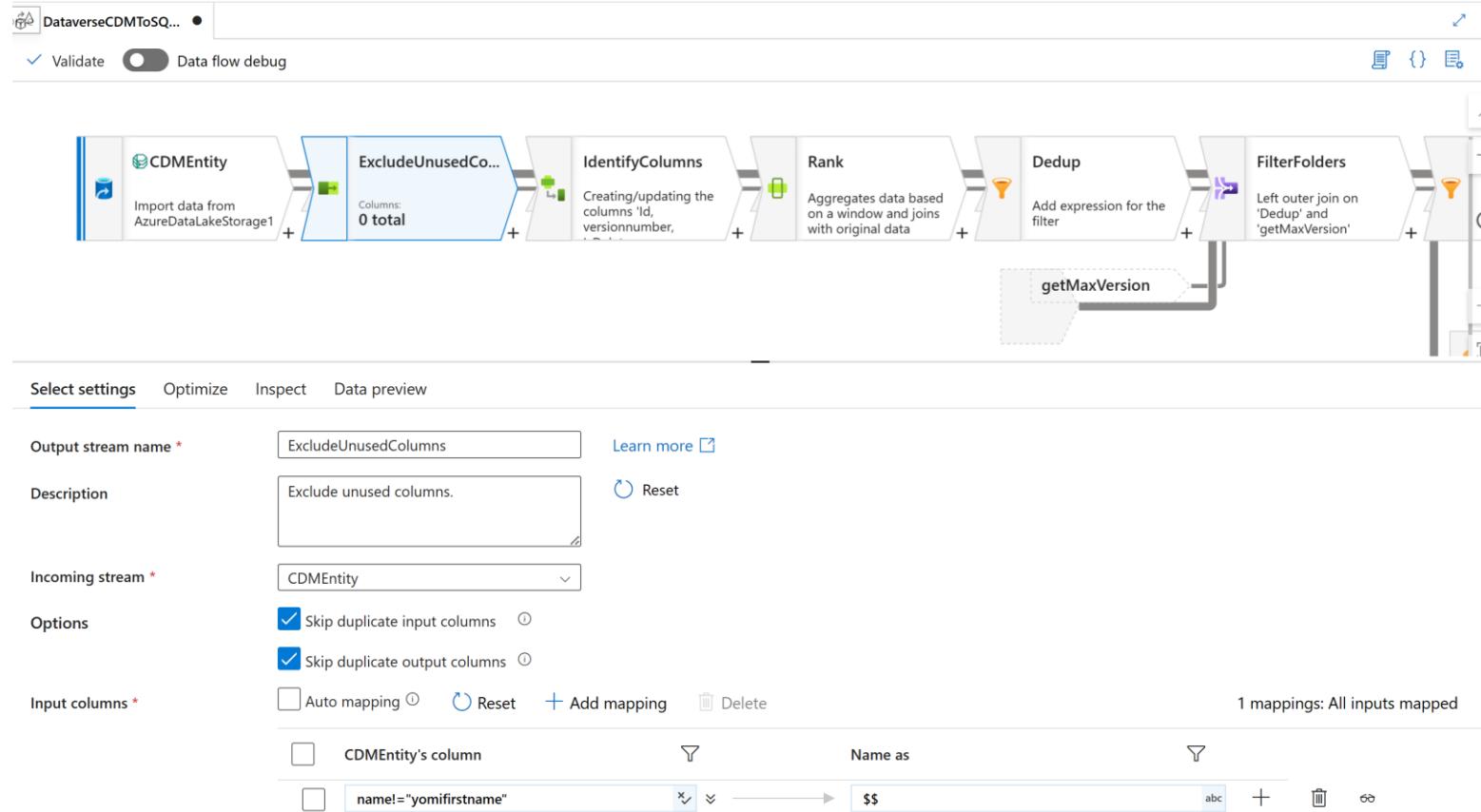
```
Declare @AlterTableAlterColumnDDL nvarchar(max) = (SELECT  
STRING_AGG(CONVERT(NVARCHAR(max), + 'alter table ' + @entity + ' ALTER COLUMN ' + '[' + name + '] ' + datatypeSQL), ';')  
from @metadata m  
left join INFORMATION_SCHEMA.COLUMNS c on  
c.TABLE_NAME = @entity and TABLE_SCHEMA = 'dbo'  
and c.COLUMN_NAME = name COLLATE DATABASE_DEFAULT  
where c.COLUMN_NAME is not null  
and (c.Data_type != datatype COLLATE DATABASE_DEFAULT or c.CHARACTER_MAXIMUM_LENGTH != maxLength or NUMERIC_PRECISION != precision or  
NUMERIC_SCALE != scale)  
);
```

*This issue will be resolved in upcoming version.

Unable to create a row greater than the allowable maximum row size (8K)

Root Cause: <https://learn.microsoft.com/en-us/power-platform/admin/replicate-data-microsoft-azure-sql-database#unable-to-create-a-row-greater-than-the-allowable-maximum-row-size-8k>

Resolution: Exclude unused columns by updating the Data Flow to include Schema modifier - Select and excluding unused columns.



Warning: The table "contact" has been created, but its maximum row size exceeds the allowed maximum of 8060 bytes. INSERT or UPDATE to this table will fail if the resulting row exceeds the size limit. The statement has been terminated.

Root Cause: <https://learn.microsoft.com/en-us/power-platform/admin/replicate-data-microsoft-azure-sql-database#unable-to-create-a-row-greater-than-the-allowable-maximum-row-size-8k>

Resolution: Exclude unused columns from table definition & Data Flow.

Step 1: Exclude unused columns by modifying create table script and adding NOT IN with list of columns to be excluded to create table and add columns

```
DECLARE @CreateTableDDL NVARCHAR(max) = (SELECT
'IF OBJECT_ID(''' + @completeDestinationTableName + ''', ''U'') IS NULL
CREATE TABLE ' + @completeDestinationTableName + ' (' + STRING_AGG(CONVERT(NVARCHAR(max), + '[' + name + '] ' +
datatypeSQL), ',') +
', CONSTRAINT ' + @destinationTableName + '_ID PRIMARY KEY CLUSTERED (Id)' + ')'
FROM @metadata m
WHERE m.Name NOT IN
('yomilastname', 'yomifirstname', 'yomifullname', 'preferredsystemuseridyominame', 'modifiedbyexternalpartyyominame'
, 'parentcustomeridyominame', 'owneridyominame'));
```

```
EXECUTE sp_executesql @CreateTableDDL;
```

```
-- add Columns
DECLARE @AlterTableAddColumnDDL NVARCHAR(max) = (SELECT
'ALTER TABLE ' + @completeDestinationTableName + ' ADD ' + STRING_AGG(CONVERT(NVARCHAR(max), + '[' + name + '] ' +
datatypeSQL), ',')
FROM @metadata m
left join INFORMATION_SCHEMA.COLUMNS c on
c.TABLE_NAME = @destinationTableName and TABLE_SCHEMA = @schemaName
and c.COLUMN_NAME = m.Name
WHERE c.COLUMN_NAME is null
AND m.Name NOT IN
('yomilastname', 'yomifirstname', 'yomifullname', 'preferredsystemuseridyominame', 'modifiedbyexternalpartyyominame'
, 'parentcustomeridyominame', 'owneridyominame'));
```

*Sample

Warning: The table "contact" has been created, but its maximum row size exceeds the allowed maximum of 8060 bytes. INSERT or UPDATE to this table will fail if the resulting row exceeds the size limit. The statement has been terminated (contd.).

Root Cause: <https://learn.microsoft.com/en-us/power-platform/admin/replicate-data-microsoft-azure-sql-database#unable-to-create-a-row-greater-than-the-allowable-maximum-row-size-8k>

Resolution: Exclude unused columns from table definition & Data Flow.

Step 2: Exclude unused columns by updating the Data Flow to include Schema modifier - Select and excluding unused columns.

The screenshot shows the Microsoft Power BI Data Flow editor interface. At the top, there is a visual representation of a data flow pipeline consisting of several components connected by arrows: CDMEntity (Import data from AzureDataLakeStorage1) → RemoveFewColumns (Column: 0 total) → IdentifyColumns (Creating/updating the column's versionnumber.) → Rank (Aggregates data based on a window and joins with original data) → Dedup (Add expression for the filter) → FilterFolders (Left outer join on 'Dedup' and 'getMaxVersion') → FilterMaxVersion (Filtering rows using expressions on column 'versionnumber.') → getMaxVersion (Final output). Below this, the 'Select settings' tab is active, showing the following configuration:

- Output stream name:** RemoveFewColumns
- Description:** Add columns to rename
- Incoming stream:** CDMEntity
- Options:** Skip duplicate input columns (checked), Skip duplicate output columns (checked)
- Input columns:** 1 mappings: All input. A table shows the mapping: CDMEntity's column (name="yomilastname"&&name!="yomifirstname") is mapped to Name as (\$\$).

Dataflow expression b

RemoveFewColumns

Matching condition

```
name!="yomilastname"&&name!="yomifirstname"&&name!="yomifullname"&&name!="preferredsystemuseridyominame"&&name!="modifiedbyexternalpartyyominame"&&name!="parentcustomeridyominame"&&name!="owneridyominame"
```

Use regular expression **Save**

Output column name expression

```
$$
```

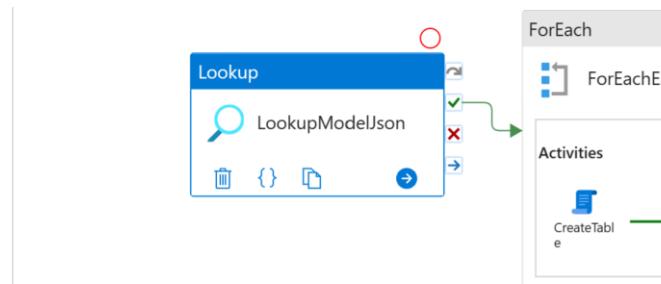
*Sample

Occasional error FileSystem: 'dataverse-xyz-orgabc'. Path: '<filename>/model.json'. ErrorCode: 'PathNotFound'. Message: 'The specified path does not exist.'

Root Cause: Missing Retry in LookupModel

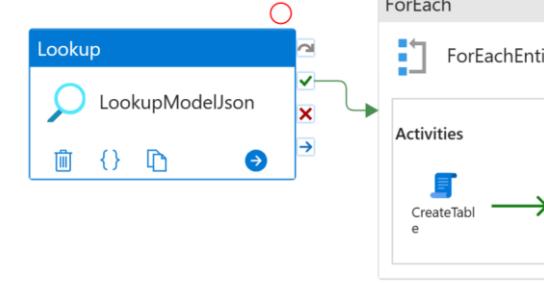
Resolution: Use appropriate value for Retry & Timeout.

Before:



General	Settings	User properties
Name *	LookupModelJson	
Description	Read model.json file to retrieve metadata	
Timeout ⓘ	7.00:00:00	
Retry ⓘ	0	
Retry interval (sec) ⓘ	30	
Secure output ⓘ	<input type="checkbox"/>	
Secure input ⓘ	<input type="checkbox"/>	

After:



General	Settings	User properties
Name *	LookupModelJson	
Description	Read model.json file to retrieve metadata	
Timeout ⓘ	0.00:05:00	
Retry ⓘ	3	
Retry interval (sec) ⓘ	30	
Secure output ⓘ	<input type="checkbox"/>	
Secure input ⓘ	<input type="checkbox"/>	

Known Limitation

Known Limitation

- Large model.json can cause failure <https://docs.microsoft.com/en-us/azure/data-factory/control-flow-lookup-activity#supported-capabilities>

Work around:

- Use separate Synapse Links for different set of entities/tables. ADF pipelines should point to same destination Azure SQL DB.
- Copy model.json to SQL DB.
<https://github.com/MicrosoftDocs/powerapps-docs/issues/3625>

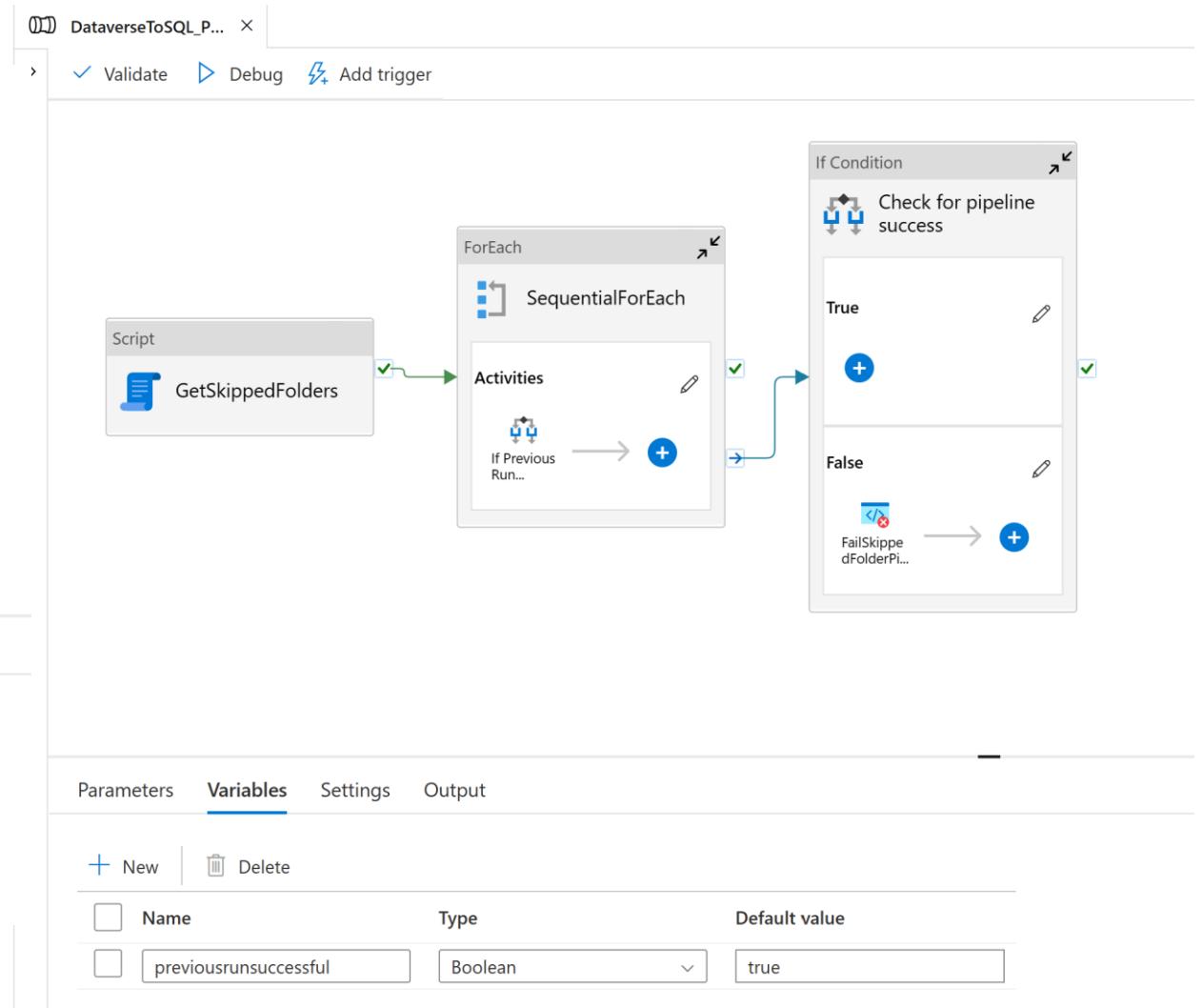
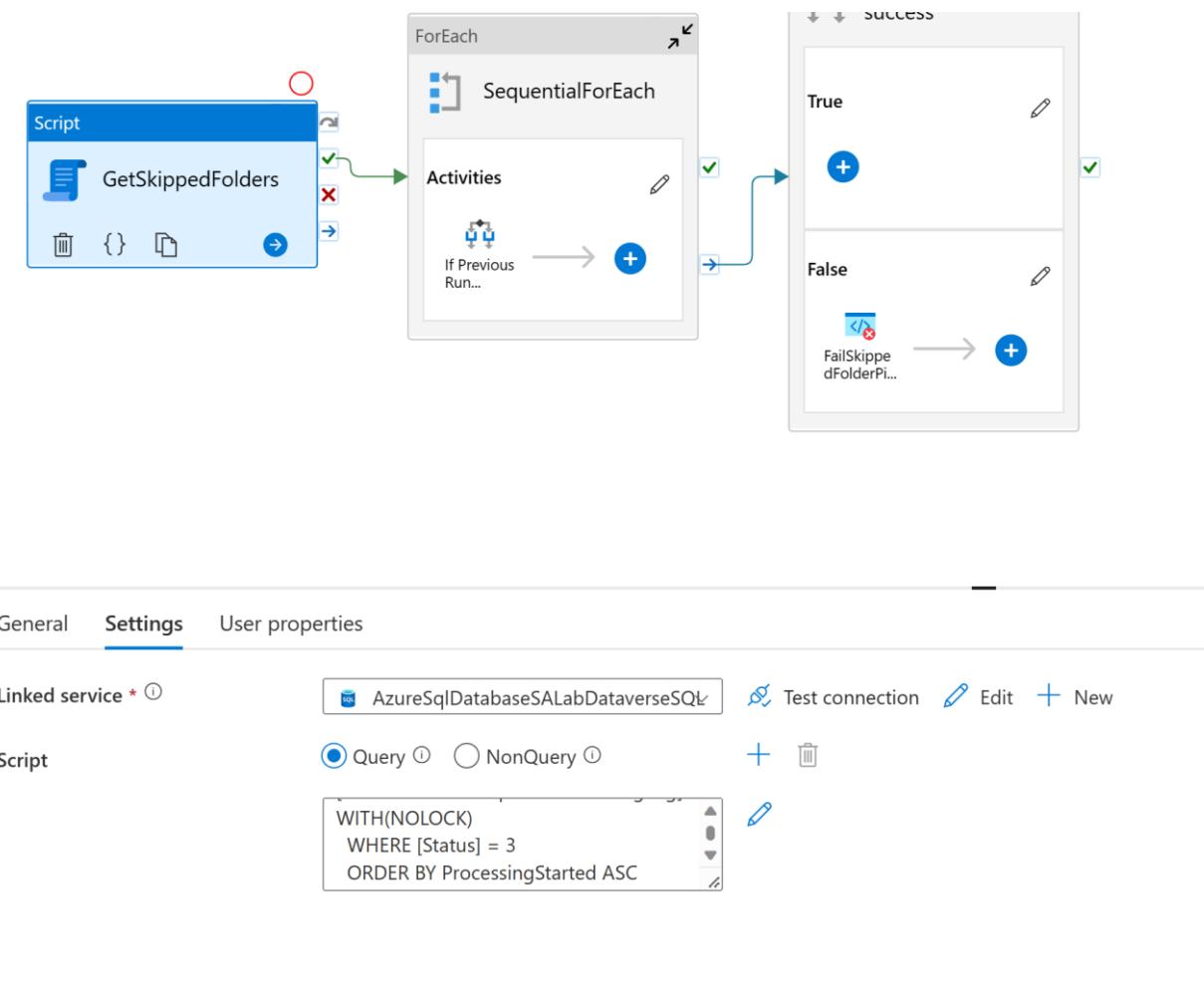
Known Limitation

- Certain multiline (for example large JSON, HTML) text might not get parsed correctly by the CDM connector. IR's geo (Spark pool settings) might influence this.
Please validate that data from entity/table like email, workflow etc. are getting copied to destination Azure SQL DB

Appendix

Sample Pipeline to process Skipped Folders

Option 1: Create a new pipeline to read the skipped folders from log table, process them chronologically & sequentially



Sample Pipeline to process Skipped Folders

Setting for the Sequential ForEach

The screenshot shows the Azure Data Factory pipeline editor with a pipeline named "DataverseToSQL_P...". The pipeline consists of a "GetSkippedFolders" activity followed by a "ForEach" activity. The "ForEach" activity is set to "Sequential" (highlighted with a red box) and contains one activity: "Check for pipeline success". An "If Condition" branch leads to a "True" path (also highlighted with a red box) which contains the expression "@activity('GetSkippedFolders').output.resultSets[0].rows". The "False" path contains the "FailSkippedFolderPipeline" activity. Below the pipeline, the "Settings" tab is selected, showing the "Sequential" setting is checked. The "Items" field contains the expression "@activity('GetSkippedFolders').output....". On the right, the "Pipeline expression builder" pane shows the expression being built.

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
@activity('GetSkippedFolders').output.resultSets[0].rows
```

Clear contents

Activity outputs Parameters System variables Functions Variables

Search

Execute DataverseToSQL
Execute DataverseToSQL activity output

GetSkippedFolders
GetSkippedFolders activity output

UpdateProcessingLogAsSuccessful
UpdateProcessingLogAsSuccessful activity output

General **Settings** Activities (1) User properties

Sequential

Items @activity('GetSkippedFolders').output....

OK Cancel

Sample Pipeline to process Skipped Folders

Check for success of previous folder since ForEach doesn't break if there an error

The screenshot shows the Azure Data Factory pipeline editor. On the left, under the 'Activities' tab, there is an 'If Previous Run Successful' activity. This activity has two branches: 'True' and 'False'. The 'True' branch contains an 'Execute Datavers...' activity followed by an 'UpdateProcessingLo...' activity. The 'False' branch contains a placeholder '+' icon. Below the main activity, there is a table with columns 'Case' and 'Activity'. The 'True' case has two activities: 'Execute Datavers...' and 'UpdateProcessing...'. At the bottom, it says '3 Activities'.

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
@bool(variables('previousrunsuccessful'))
```

ForEach iterator [Activity outputs](#) [Parameters](#) [System variables](#) [Functions](#) [Var](#)

[Search](#)

SequentialForEach
Current item

General Activities (3) User properties

Expression ⓘ @bool(variables('previousrunsuccessful'))

Case	Activity
True	Execute Datavers... UpdateProcessing...

3 Activities

OK Cancel

Sample Pipeline to process Skipped Folders

IF – True

DataverseToSQL_P... X

Validate Debug Add trigger

DataverseToSQL_ProcessSkippedRows_V2 > SequentialForEach > If Previous Run Successful > True activities

General Settings User properties

Invoked pipeline *

DataverseToSQL_withoutDF Open New

Wait on completion

Parameters

Name	Type	Value	Default value
container	string	@item().container	
folder	string	@item().folder	

Environment variables

Name	Type	Value
container	string	@item().container
folder	string	@item().folder

Script

UpdateProcessingLogAsSuccessful

Execute Pipeline

Execute DataverseToSQL

Set variable

(X) Set previous run status as false

```
graph LR; Start(( )) --> S1[SequentialForEach]; S1 --> S2[If Previous Run Successful]; S2 -- True --> E1[Execute Pipeline]; E1 --> S3[Set variable]; S3 --> E2[Execute Pipeline]; E2 --> S4[Script]; S4 --> End(( ));
```

The screenshot shows the 'Settings' tab of a Microsoft Power Automate pipeline named 'DataverseToSQL_P...'. The 'Invoked pipeline' dropdown is set to 'DataverseToSQL_withoutDF'. The 'Wait on completion' checkbox is checked. Below the tabs, there are two tables: 'Parameters' and 'Environment variables'. The 'Parameters' table has two rows: 'container' (Type: string, Value: '@item().container') and 'folder' (Type: string, Value: '@item().folder'). The 'Environment variables' table also has two rows: 'container' (Type: string, Value: '@item().container') and 'folder' (Type: string, Value: '@item().folder'). Above the settings, the pipeline's logic is visible: it starts with a 'SequentialForEach' step, followed by an 'If Previous Run Successful' step. The 'True' condition leads to an 'Execute Pipeline' step (invoking 'DataverseToSQL'), which then leads to a 'Set variable' step ('Set previous run status as false'). This is followed by another 'Execute Pipeline' step (invoking 'DataverseToSQL'), which finally leads to a 'Script' step ('UpdateProcessingLogAsSuccessful').

Sample Pipeline to process Skipped Folders

Script component to update the folder to success

> Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
@concat('UPDATE dbo.DataVerseToSQLPipelineProcessingLog
SET
    Status = 1
WHERE Container = ', item().container, ' AND Folder = ', item().folder, '')
```

[Clear contents](#)

[Activity outputs](#) [Parameters](#) [System variables](#) [Functions](#) [Variables](#)

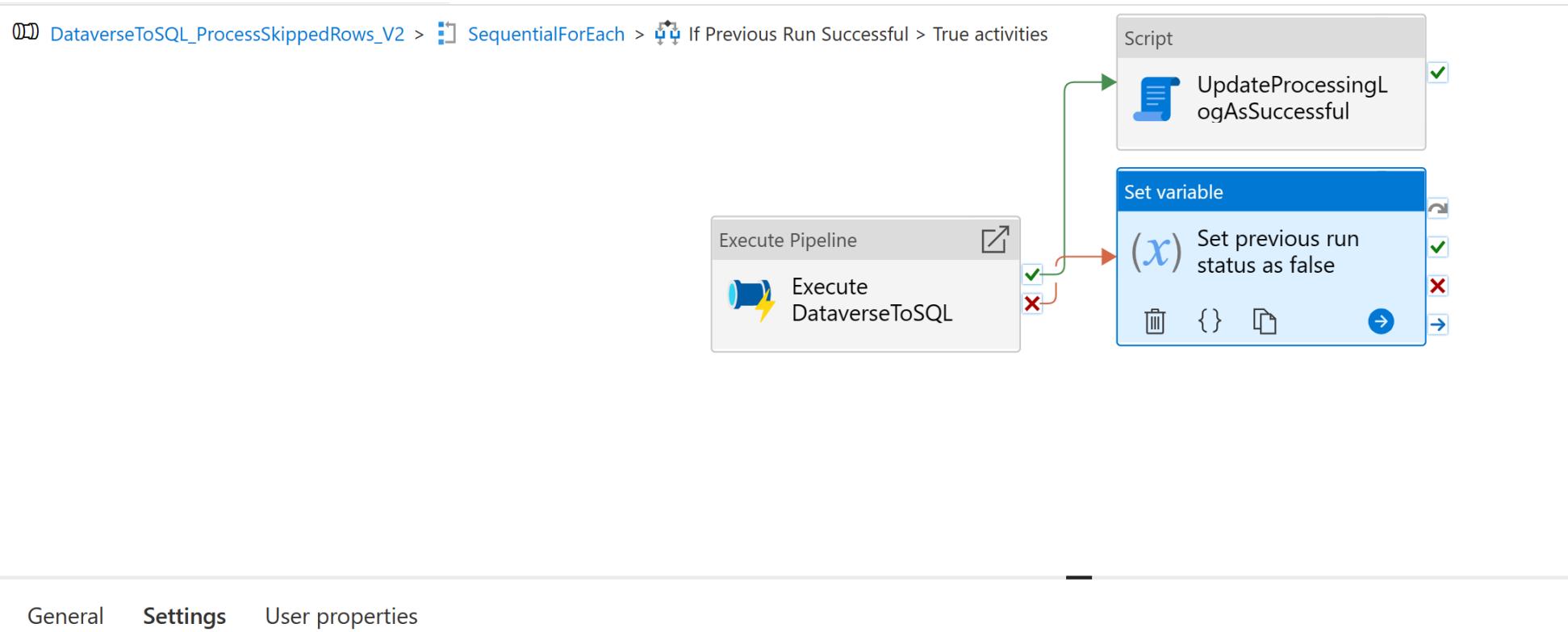
[Search](#)

Execute DataverseToSQL
Execute DataverseToSQL activity output

GetSkippedFolders
GetSkippedFolders activity output

Sample Pipeline to process Skipped Folders

Set variable if there is any pipeline failure.



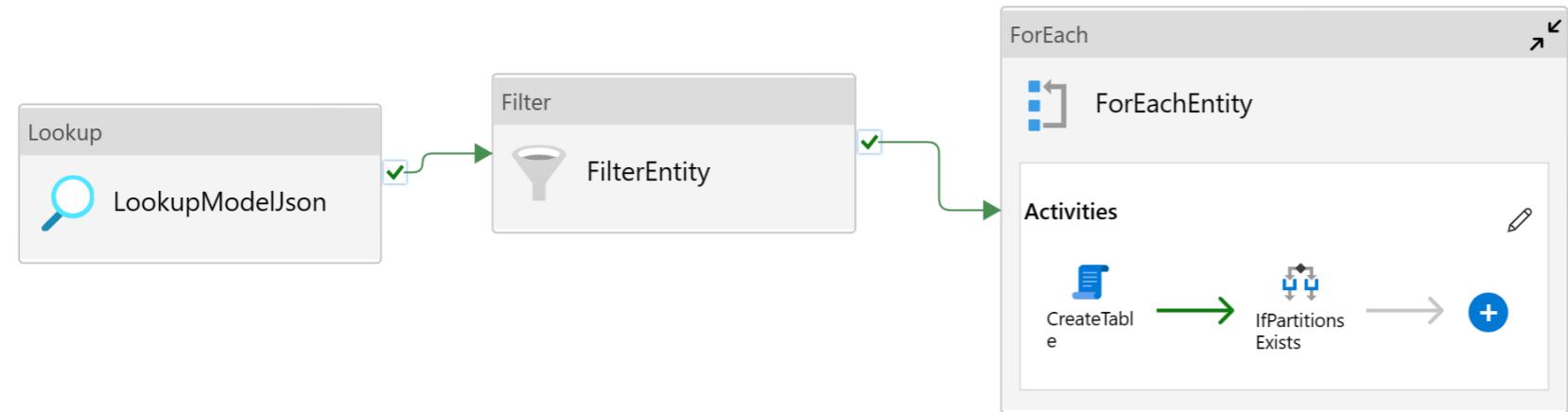
General **Settings** User properties

Name * previousrununsuccessful + New

Value @bool(0)

Sample Pipeline to process single entity

Filter LookupModelJson output



Sample Pipeline to process single entity

Filter Items

The screenshot shows the Azure Data Factory Pipeline expression builder interface. On the left, there's a preview of the pipeline flow: a 'Lookup' activity followed by a 'Filter' activity. The 'Settings' tab is selected for the 'Filter' activity. In the 'Items' section, the expression is set to `@activity('LookupModelJson').output.firstRow.entities`. In the 'Condition' section, the expression is set to `@equals(toLower(item().name), 's')`. On the right, the 'Pipeline expression builder' pane displays the expression `@activity('LookupModelJson').output.firstRow.entities` and lists the available activity outputs: `LookupModelJson` (LookupModelJson activity output) and `LookupModelJson first row` (Data of the first row).

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
@activity('LookupModelJson').output.firstRow.entities
```

Clear contents

Activity outputs Parameters System variables Functions Variables

Search

LookupModelJson
LookupModelJson activity output

LookupModelJson first row
Data of the first row

General Settings User properties

Items Condition

```
@activity('LookupModelJson').out
```

```
@equals(toLower(item().name), 's')
```

Sample Pipeline to process single entity

Filter Condition

The screenshot shows the Azure Data Factory Pipeline expression builder interface. On the left, the pipeline structure is visible with a 'Lookup' activity followed by a 'Filter' activity. The 'Filter' activity has a green checkmark icon indicating it is active. On the right, the 'Pipeline expression builder' window is open, displaying the expression: `@equals(toLower(item().name), 'email')`. Below the expression, there are tabs for 'Activity outputs', 'Parameters', 'System variables', 'Functions', and 'Variables'. Under 'Activity outputs', two items are listed: 'LookupModelJson' (LookupModelJson activity output) and 'LookupModelJson first row' (Data of the first row). The 'Settings' tab is selected in the pipeline editor.

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#).

```
@equals(toLower(item().name), 'email')
```

Clear contents

Activity outputs Parameters System variables Functions Variables

Search

LookupModelJson
LookupModelJson activity output

LookupModelJson first row
Data of the first row

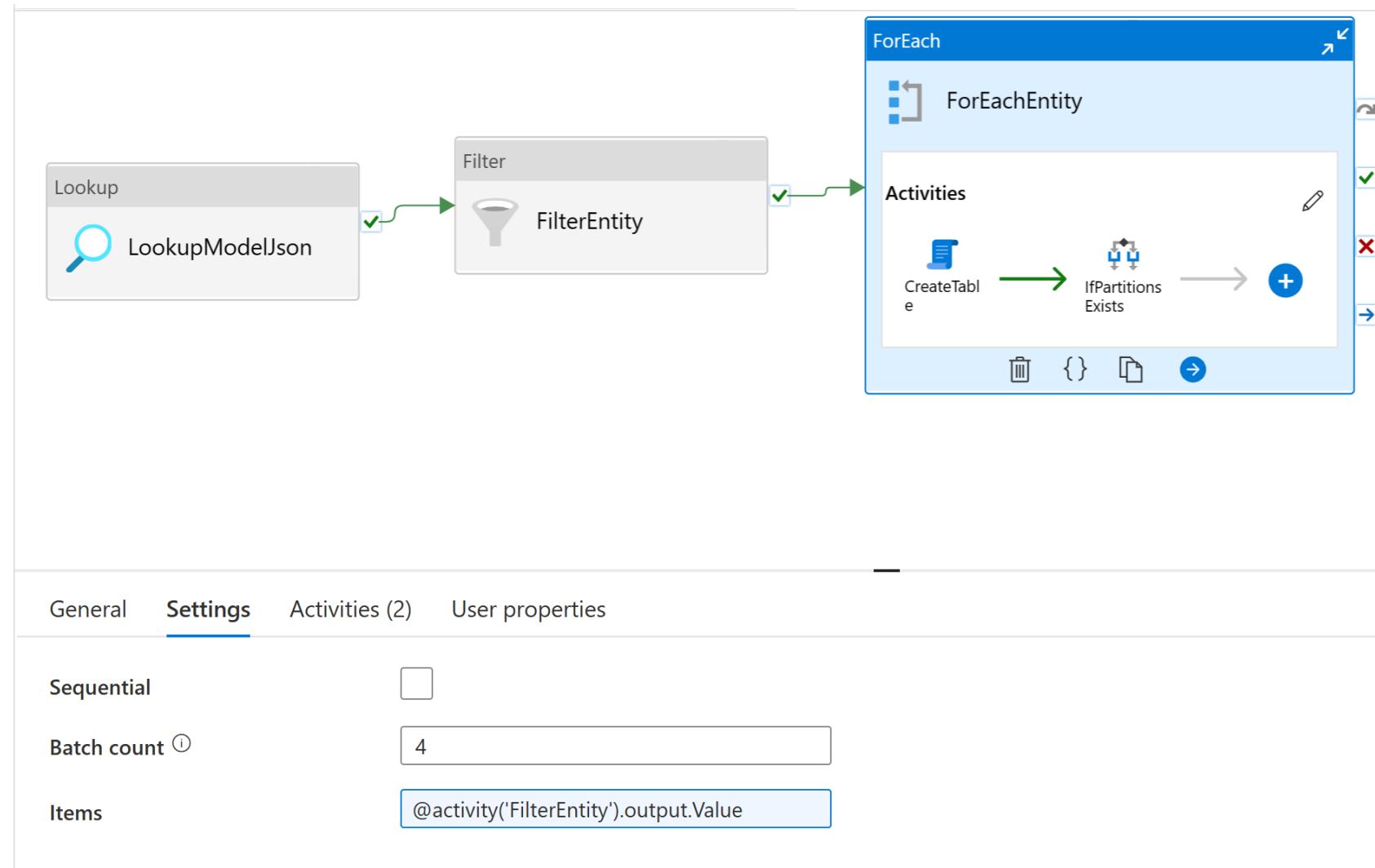
General Settings User properties

Items @activity('LookupModelJson').out

Condition @equals(toLower(item().name), 'email')

Sample Pipeline to process single entity

ForEach Items



Sample Pipeline update to Exclude entities without partitions

Filter Condition

What you think, share your feedback in this quick [survey](#) about Azure Data Factory

Data Factory Validate all Publish all

Dataflow: DataverseToSQL_O... X

Validate Debug Add trigger Data flow debug

The screenshot shows a pipeline named "DataverseToSQL_O...". It starts with a "LookupModelJson" activity, followed by a "Filter" activity with the condition "FilterEntityWithEmptyPartitions". The output of the filter activity feeds into an "ForEach" activity. Inside the "ForEach" loop, there is a "CreateTable" activity, which then connects to a "DataverseToSQL_Online" activity.

Pipeline expression builder

Add dynamic content below using any combination of [expressions](#), [functions](#) and [system variables](#):

```
@greater(length(item().partitions),0)
```

Activity outputs Parameters System variables Functions Variables

Search

LookupModelJson
LookupModelJson activity output

LookupModelJson first row
Data of the first row

General **Settings** User properties

Items: @activity('LookupModelJson').output.f...

Condition: @greater(length(item().partitions),0)

Sample Pipeline update to Exclude entities without partitions

ForEach Items

The screenshot shows the Azure Data Factory interface with a pipeline named "DataverseToSQL_O...".

Pipeline Expression Builder:

```
@activity('FilterEntityWithEmptyPartitions').output.Value
```

Activity Details:

- General Tab:** Sequential checkbox is unchecked. Batch count is set to 4. Items are defined as `@activity('FilterEntityWithEmptyPartitions').output.Value`.
- Settings Tab:** Sequential checkbox is checked.
- Activities Tab:** Contains three activities: "ForEachEntity" (selected), "CreateTable", and "DataverseToSQL_Online".

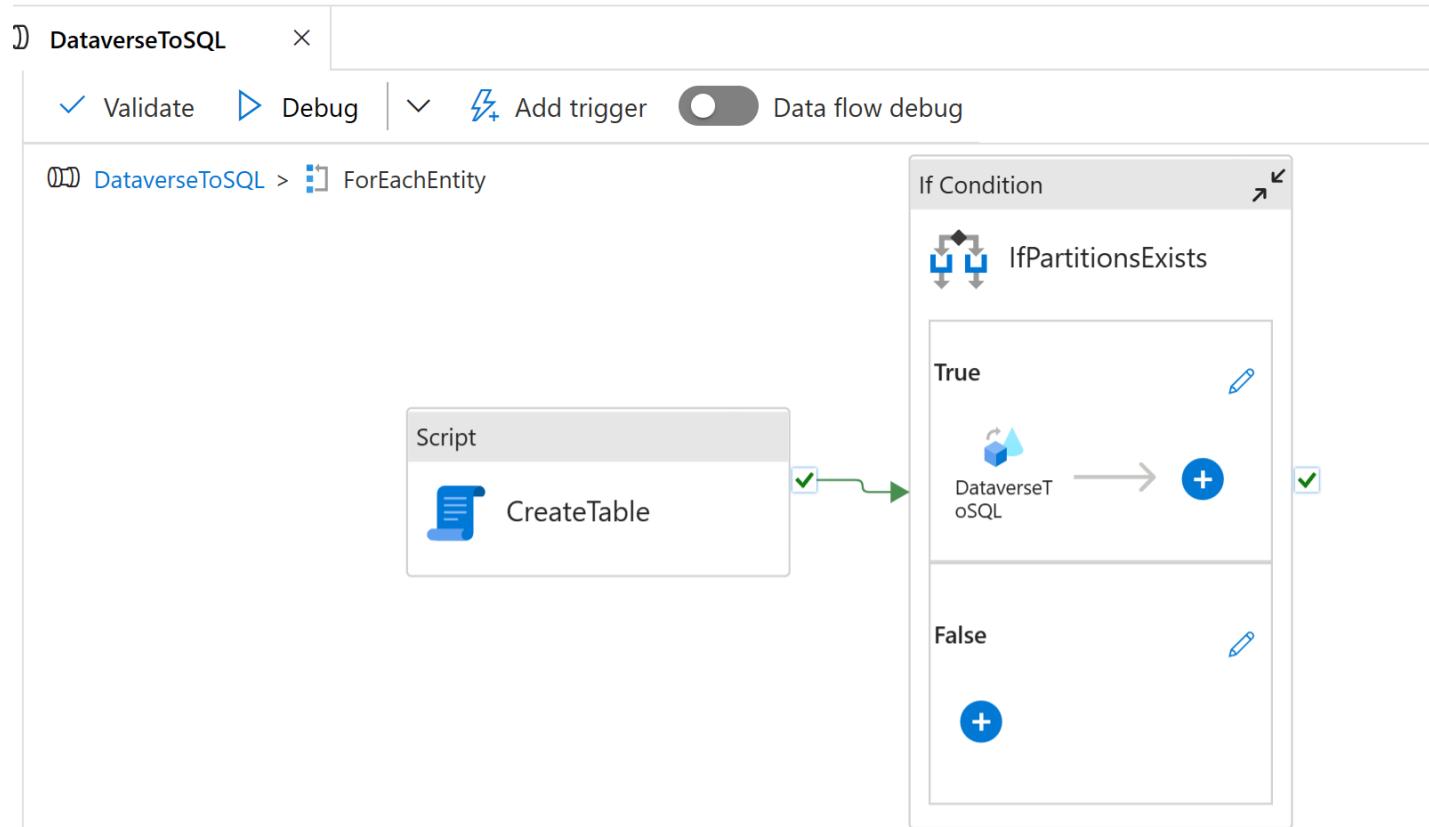
ForEach Entity Activity:

- Items:** `ookupModelJson`
- Filter:** `FilterEntityWithEmptyPartitions`
- Activities:** `CreateTable` followed by `DataverseToSQL_Online`.

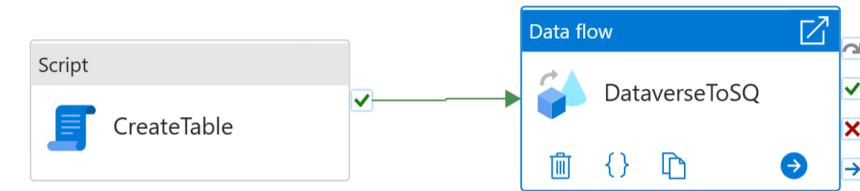
Sample Pipeline update to Exclude entities without partitions

Inside ForEach – Remove If Condition

Before



After



Sample Pipeline update to enable custom logging

Sample script to create Log table for each entity/table

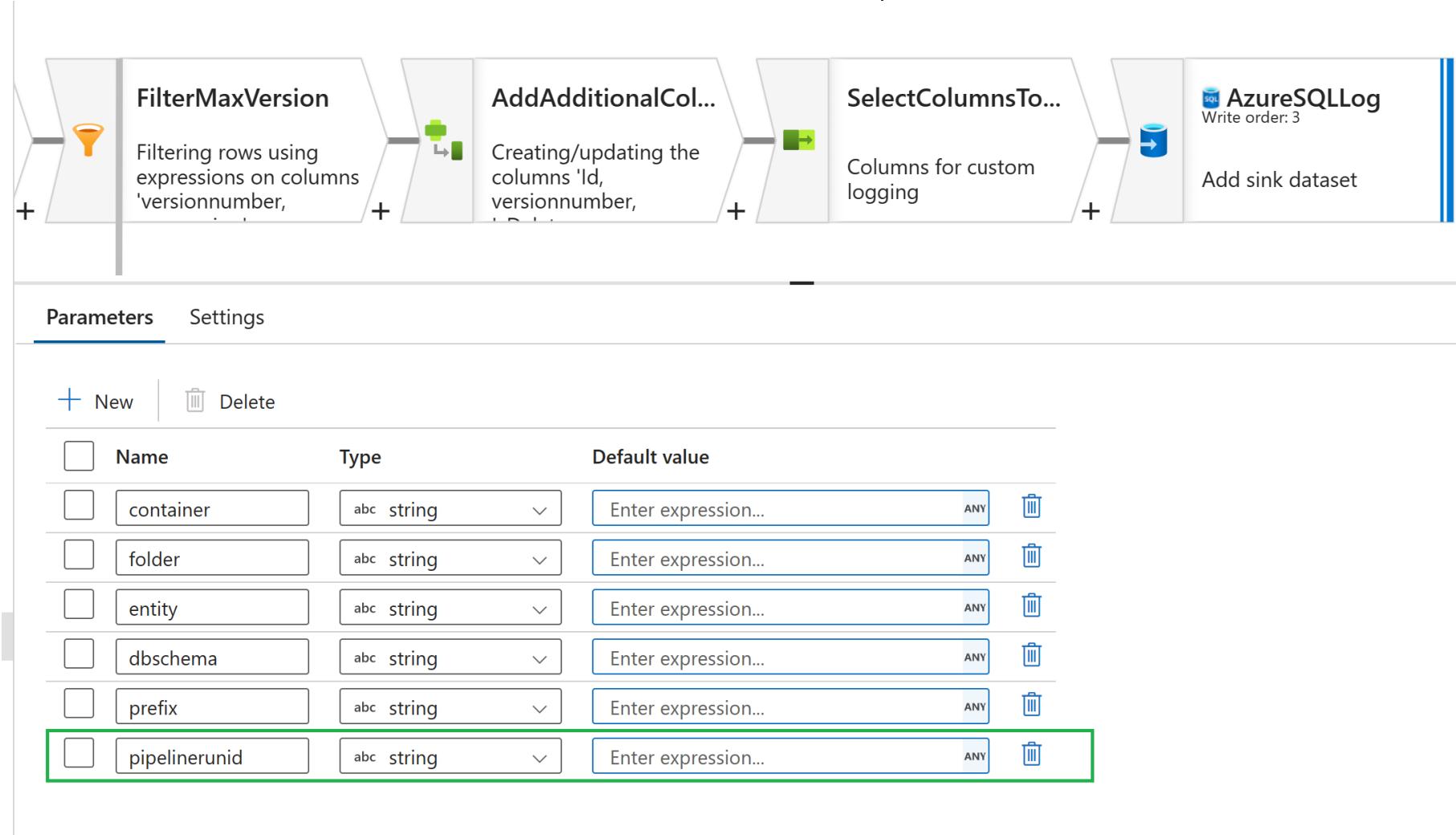
```
DECLARE @CreateEntityLogTableDDL NVARCHAR(max) =
'IF OBJECT_ID('' + @schemaName + '.' + @tablePrefix + @entity + 'log'', ''U'') IS NULL
CREATE TABLE ' + @schemaName + '.' + '[' + @tablePrefix + @entity + 'log](
[Id] [bigint] IDENTITY(1,1) NOT NULL,
[LogDate] [datetime2](7) NOT NULL DEFAULT(GETUTCDATE()),
[RecordId] [nvarchar](64) NOT NULL,
[SinkModifiedOn] [datetime] NOT NULL,
[VersionNumber] [bigint] NOT NULL,
[MaxVersion] [bigint] NOT NULL,
[Container] [nvarchar](100) NOT NULL,
[Folder] [nvarchar](100) NOT NULL,
[PipelineRunId] [nvarchar](50) NOT NULL,
[IsDelete] [bit] NULL
PRIMARY KEY CLUSTERED
(
[Id] ASC
)
) ON [PRIMARY]';

EXECUTE sp_executesql @CreateEntityLogTableDDL;
```

Use desired schema, table prefix and create one log table for each entity of Synapse Link in separate log database.

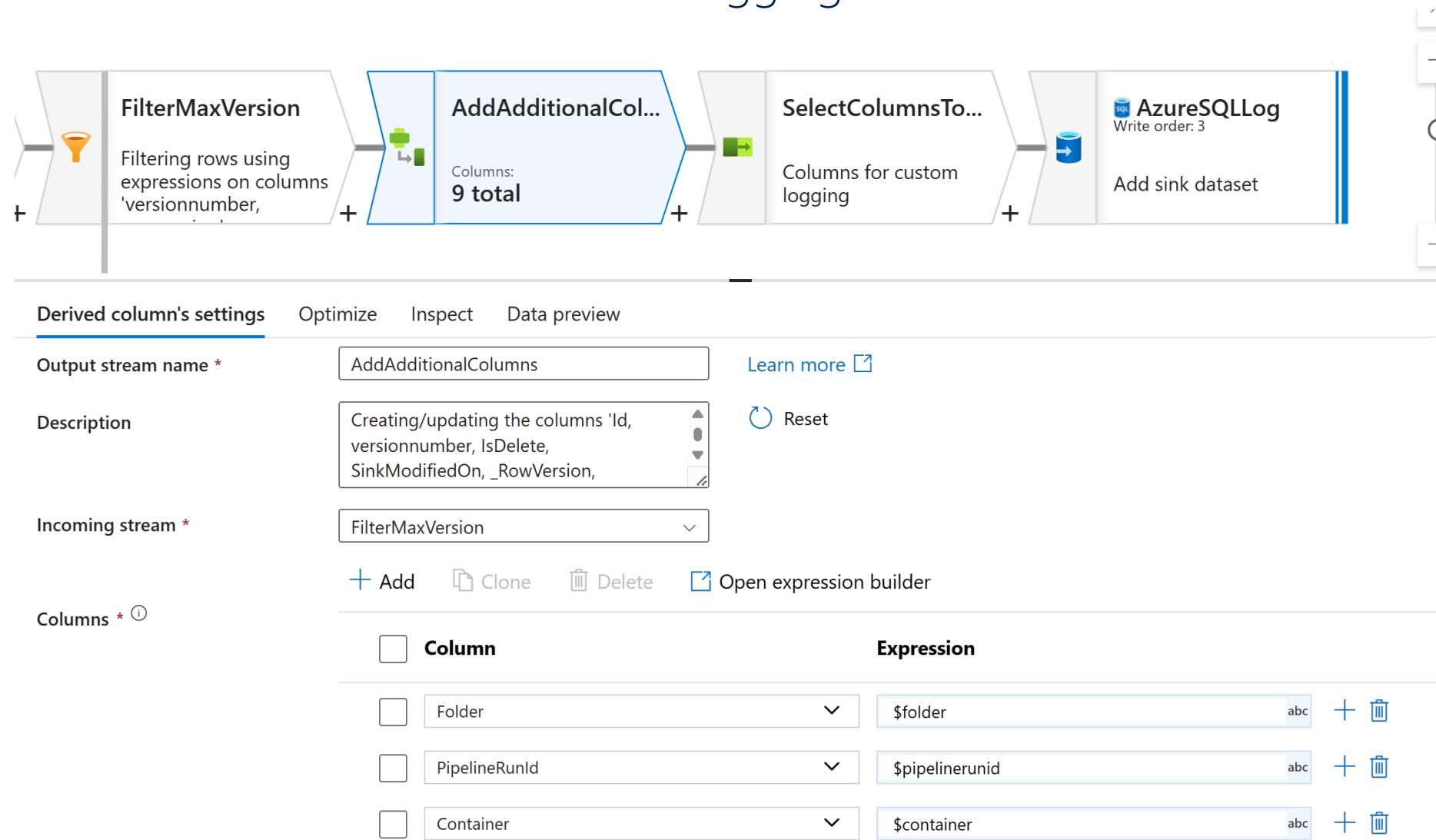
Sample Pipeline update to enable custom logging

Inside Data Flow – Add another variable for Pipeline Run Id



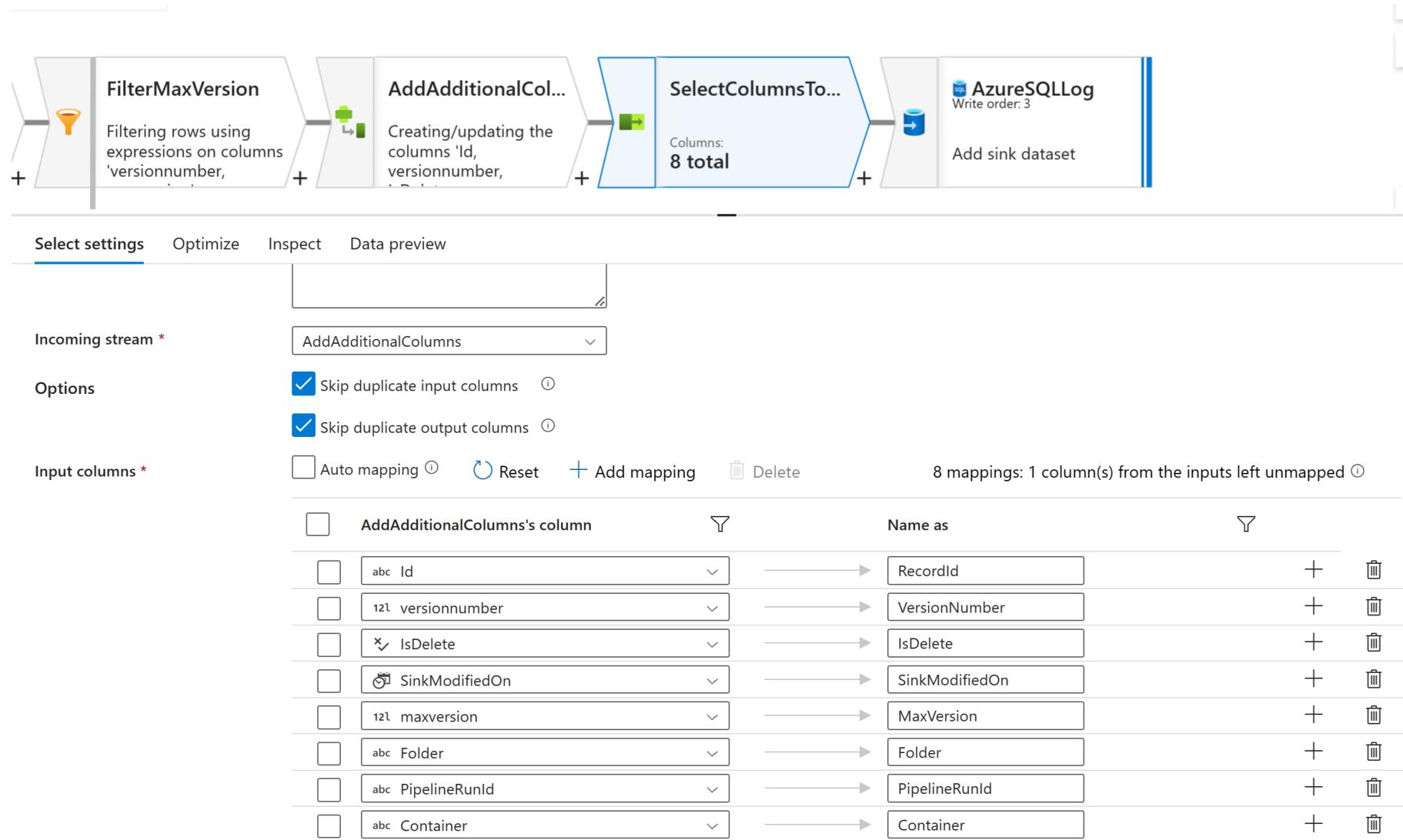
Sample Pipeline update to enable custom logging

Inside Data Flow – Add another branch for logging



Sample Pipeline update to enable custom logging

Inside Data Flow – Add another branch for logging



Sample Pipeline update to enable custom logging

Inside Data Flow – Add another branch for logging



Sink Settings Errors Mapping Optimize Inspect Data preview

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

Description Add sink dataset [Reset](#)

Incoming stream * SelectColumnsToBeLogged

Sink type * Dataset Inline Cache

Inline dataset type * Azure SQL Database

Linked service * AzureSqlDatabase_DataVerse01 [Test connection](#) [Edit](#) [New](#)

Options Allow schema drift [①](#) Validate schema [①](#)

Please use separate database (linked service) for logging.

Sample Pipeline update to enable custom logging

Inside Data Flow – Add another branch for logging



Sink **Settings** Errors Mapping Optimize Inspect Data preview

Schema name * \$dbschema abc Refresh

Table name * \$prefix + \$entity + 'log' abc

Table action None Recreate table Truncate table

Update method Allow insert
 Allow delete
 Allow upsert
 Allow update

Use TempDB

Pre SQL scripts List of scripts Custom expression ⓘ

Sample Pipeline update to enable custom logging

Inside Data Flow – Add Sink Ordering to ensure Logging happens at the end

The screenshot shows the 'Settings' tab of the Azure Data Flow pipeline configuration. Under the 'Custom sink ordering' section, there are three entries:

Sink name	Write order
AzureSQLDB	1
AzureSQLDBDeleteLog	2
AzureSQLLog	3

The sinks listed are AzureSQLDB, AzureSQLDBDeleteLog, and AzureSQLLog. The write order is set to 1 for AzureSQLDB, 2 for AzureSQLDBDeleteLog, and 3 for AzureSQLLog. This ordering ensures that the log sink is the last one in the pipeline.

Sample Pipeline update to enable custom logging

Inside ForEach – Data Flow -> Map new parameter pipelinerunid as pipeline expression

The screenshot shows the 'Data flow' configuration screen in Power Automate. At the top, there's a preview window for a 'DataverseToSQL' step. Below it, the 'Parameters' tab is selected. A new parameter, 'pipelinerunid', has been added to the list, highlighted with a green border. The 'Value' for this parameter is set to the pipeline expression '@pipeline().RunId'. The table also includes other parameters like 'container', 'folder', 'entity', 'dbschema', and 'prefix'.

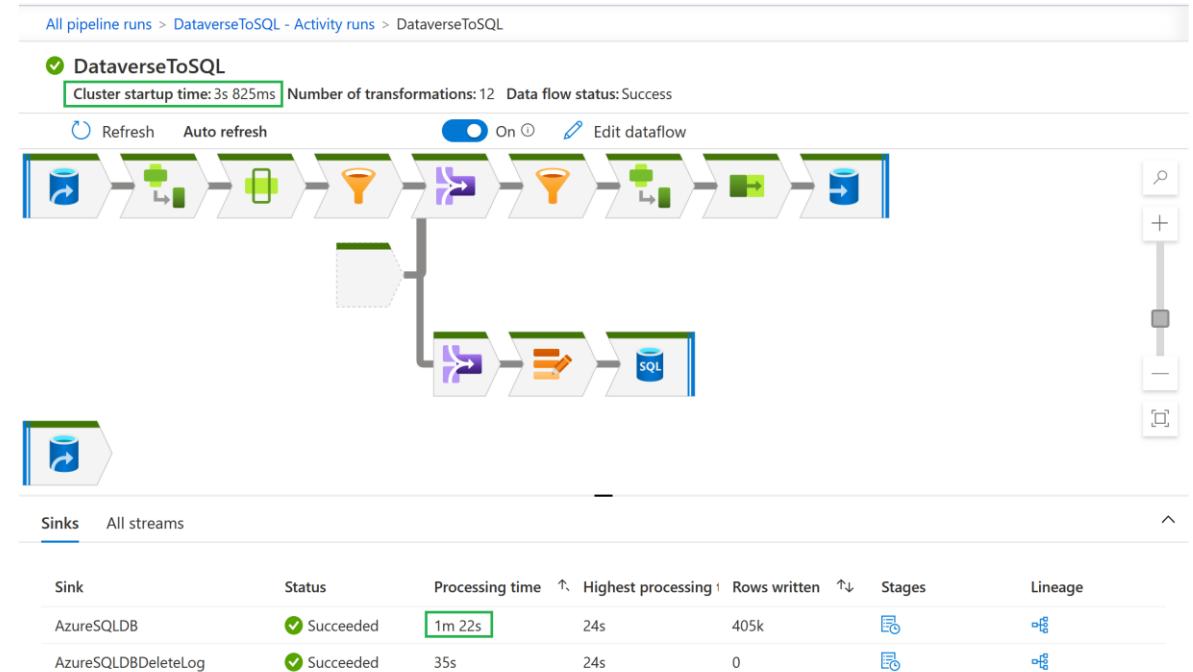
Name	Value	Type	Expression ⓘ
container	@pipeline().parameters.container	string	<input type="text"/>
folder	@pipeline().parameters.folder	string	<input type="text"/>
entity	@item().name	string	<input type="text"/>
dbschema	@variables('dbschema')	string	<input type="text"/>
prefix	@variables('prefix')	string	<input type="text"/>
pipelinerunid	@pipeline().RunId	string	<input type="text"/>

Sample

Default IR



Custom IR with higher TTL & size





© 2023 Microsoft Corporation. All rights reserved. Microsoft, Microsoft Dynamics, Office 365, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION