



Loading Data using Pipelines – Storage Account

Pipelines in Azure Synapse Analytics

A **pipeline in Azure Synapse Analytics** is a collection of activities that automate data movement and transformation workflows. Pipelines are part of **Azure Synapse Pipelines**, which offer **data integration capabilities** to orchestrate Extract, Transform, and Load (ETL) or Extract, Load, and Transform (ELT) processes. These pipelines can manage complex workflows, integrate various data sources, and process large-scale data effectively.

What is the Copy Data Tool in Azure Synapse?

The **Copy Data Tool** is a feature in Azure Synapse that simplifies the process of copying data from one source to a destination. It is designed for **data migration, ingestion, and integration** tasks, enabling users to efficiently move data between different storage systems and databases.

Key Features of the Copy Data Tool

1. Ease of Use:

- Provides a simple, wizard-based interface for creating data copy pipelines without requiring deep technical knowledge.

2. Support for Multiple Data Sources:

- Supports copying data from diverse sources, including:
 - Azure Blob Storage
 - Azure Data Lake Storage
 - SQL Server
 - Azure SQL Database
 - REST APIs
 - SaaS applications (e.g., Salesforce)
 - On-premises systems (via integration runtime)

3. Scalability:

- Optimized for handling large-scale data ingestion with parallel processing.

4. Flexible Scheduling:

- Allows you to schedule data copy operations on a **one-time or recurring** basis.

5. Data Transformation:

- Basic transformations like column mapping and data format conversion can be applied during data movement.

6. Integration with Synapse Pipelines:

- Copy Data Tool workflows are built into Synapse Pipelines and can be extended to include additional activities.

7. Monitoring:

- Provides detailed monitoring and logging for tracking data copy progress, success, and errors.

Steps to Use the Copy Data Tool

1. Open the Copy Data Tool:

- In Azure Synapse Studio, navigate to the **Integrate** hub and select **Copy Data**.

2. Configure Source Data:

- Choose the data source from a list of supported connectors.
- Specify the connection details, such as server, database, and credentials.

3. Configure Destination:

- Select the destination where the data will be copied.
- Define storage options and target schema if applicable.

4. Define Data Mapping:

- Map columns between the source and destination.
- Optionally, perform data transformations during the mapping step.

5. Set Up Scheduling:

- Choose whether the copy operation will run immediately, on a schedule, or on-demand.

6. Run and Monitor:

- Execute the copy pipeline and monitor its progress through Synapse's monitoring tools.

Use Cases for the Copy Data Tool

1. Data Ingestion:

- Load raw data from various sources into Azure Data Lake or Dedicated SQL Pools for analytics.

2. Data Migration:

- Migrate data from on-premises systems to Azure cloud storage or databases.

3. ETL/ELT Pipelines:

- Quickly implement the **Extract** and **Load** phases of ETL or ELT workflows.

4. Backup and Archiving:

- Copy data from operational systems to long-term storage for backup or compliance purposes.

5. Hybrid Data Integration:

- Synchronize data between on-premises and cloud environments.

Benefits of the Copy Data Tool

- **Simplified Data Movement:** No-code or low-code approach to data integration.
- **Speed:** Parallel data transfer capabilities for large-scale ingestion.
- **Flexibility:** Supports a wide range of data formats and storage systems.
- **Cost-Efficient:** Automates data movement without requiring additional tools or infrastructure.

Example Pipeline Using the Copy Data Tool

Scenario: Ingest sales data from an on-premises SQL Server database into Azure Data Lake for analysis.

1. **Source:** SQL Server (on-premises).

2. **Destination:** Azure Data Lake Storage Gen2.

3. **Steps:**

- Create a linked service to SQL Server using a self-hosted integration runtime.
- Use the Copy Data Tool to define the source query and map it to a destination folder in Azure Data Lake.
- Set a daily schedule for the pipeline to run automatically.
- Monitor the pipeline to ensure successful data ingestion.

The **Copy Data Tool** is an essential feature of Azure Synapse Analytics for orchestrating and automating data integration workflows, ensuring efficient and reliable data movement across platforms.

To begin with the Lab

1. In this lab, we are going to use the pipelines to load the data into the tables. So, first, we will run the Delete command to delete the data in the table and then check that our table should be empty.

```

35  DELETE FROM PoolActivityLog
36
37  SELECT * FROM PoolActivityLog;
38

```

Results Messages

View Table Chart Export results ▾

Search

Correlationid	Operationname	Status	Eventcategory	Level	Time	Subscription	Eventinitiatedby	Resourcetype

00:00:01 Query executed successfully.

- Also, we have updated our container and here you can see that we only two parquet files.

The screenshot shows the Azure Storage Blob container named 'data'. The left sidebar includes options like Overview, Access Control (IAM), Settings, Shared access tokens, Manage ACL, Access policy, Properties, and Metadata. The main area displays a table of blobs:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
ActivityLog01.parquet	12/24/2024, 10:13:27...	Hot (Inferred)		Block blob	230.51 KIB	Available
ActivityLog02.parquet	12/24/2024, 4:51:49 ...	Hot (Inferred)		Block blob	294.49 KIB	Available

- Now, we need to go to the integrate tab, click on the Plus icon, and choose the Copy Data tool.

The screenshot shows the Synapse live interface with the 'Integrate' tab selected. The top navigation bar includes 'Synapse live', 'Validate all', and 'Publish all'. On the left, there's a vertical sidebar with icons for Home, Database, Table, Pipeline, and a redboxed icon for Copy Data tool. A search bar says 'Filter resources by name'. Below the sidebar is a list of integration tools:

- Pipeline
- Link connection
- Copy Data tool** (highlighted with a red box)
- Browse gallery
- Import from pipeline template

- In our Copy data tool, we will first choose the built-in copy task, run once now, and click on next.

Copy Data tool

The screenshot shows the 'Properties' step of the Copy Data tool wizard. On the left, a vertical navigation bar lists steps 1 through 5: Properties, Source, Destination, Settings, and Review and finish. Step 1 is highlighted with a blue checkmark. The main content area starts with a brief description: 'Use Copy Data Tool to perform a one-time or scheduled data load from 90+ data sources. Follow the wizard experience to specify your data loading settings, and let the Copy Data Tool generate the artifacts for you, including pipelines, datasets, and linked services.' Below this is a 'Properties' section with the sub-instruction: 'Select copy data task type and configure task schedule'. Under 'Task type', there are two options: 'Built-in copy task' (described as a single pipeline for smooth copying) and 'Metadata-driven copy task' (described as parameterized pipelines reading metadata from an external store). A note below states: 'You will get single pipeline to quickly copy objects from data source store to destination in a very intuitive manner.' At the bottom of the step, there's a 'Task cadence or task schedule*' section with three radio button options: 'Run once now' (selected), 'Schedule', and 'Tumbling window'.

5. Our second step is to choose our source, which is our storage account or Data Lake here will choose our container as a source object and click on next.

Copy Data tool

The screenshot shows the 'Source' step of the Copy Data tool wizard. The left navigation bar shows steps 1 through 5, with step 2 highlighted. The main content area is titled 'Source data store' and instructs the user to specify the source data store. It includes fields for 'Source type' (set to 'All'), 'Connection' (set to 'thestorageaccount1201'), and 'Integration runtime' (set to 'AutoResolveIntegrationRuntime'). Below these, there's a 'File or folder' input field containing 'data/' which is highlighted with a red box. Underneath, there are 'Options' with three checkboxes: 'Binary copy' (unchecked), 'Recursively' (checked), and 'Enable partitions discovery' (unchecked).

6. Then it will ask for the file format settings you have to choose Parquet and choose compression type to snappy. Click on next.

Copy Data tool

The screenshot shows the 'File format settings' step of the Copy Data tool wizard. The left navigation bar shows steps 1 through 5, with step 2 highlighted. The main content area has a 'File format' dropdown set to 'Parquet' and a 'Compression type' dropdown set to 'snappy' (which is highlighted with a blue box). There's also a 'Preview data' button and an 'Additional columns' section with a '+ New' button.

7. For the third step, which is the destination we will choose our Dedicated SQL Pool and then choose our existing table which is Pool Activity Log. Click on Next.

Copy Data tool

Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.

Destination type: All

Connection *: DataPool

Source	Destination
AzureBlobFSFile	dbo.PoolActivityLog

Auto-create a destination table with the source schema

8. Then for our last step, we will give a name to the pipeline and choose Bulk insert for Copy method.

Copy Data tool

Settings

Enter name and description for the copy data task, more options for data movement

Task name *: Parquet Pipeline

Task description:

Fault tolerance ⓘ: (dropdown menu)

Enable logging ⓘ: (checkbox)

Enable staging ⓘ: (checkbox)

Advanced

Copy method: Copy command ⓘ PolyBase ⓘ Bulk insert Upsert

Bulk insert table lock ⓘ: Yes No

9. We will review the pipeline and then deploy it. Once it is done then we can just click on finish.

Copy Data tool

- Properties
- Source
- Destination
- Settings
- Review and finish
- Review
- Deployment

Deployment complete

Deployment step	Status
Validating copy runtime environment	Succeeded
> Creating datasets	Succeeded
> Creating pipelines	Succeeded
> Running pipelines	Succeeded

Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close Copy Data Tool.

10. If we go to the monitor tab, we will see our pipeline whose status is succeeded.

Pipeline runs

Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run	P
Parquet Pipeline	12/24/2024, 5:29:31 PM	12/24/2024, 5:29:56 PM	25s	Manual trigger	Succeeded	Original	

11. Now if we go and run the Select command we will see the data inside the table.

21 `SELECT * FROM PoolActivityLog;`

Results Messages

View Table Chart Export results

Search

Correlationid	Operationname	Status	Eventcategory	Level	Time	Subscription	Eventinitiatedby	Resourcekey
0310b3f0-eb74...	Delete SQL con...	Succeeded	Administrative	Informational	2024-05-07T17:...	387407e5-94af...	cloudlearning4...	Microsoft.D
e02e97a6-7b45...	Update website	Succeeded	Administrative	Informational	2024-05-07T11:...	387407e5-94af...	cloudlearning4...	Microsoft.V
13c9dd2e-dda4...	Create or updat...	Succeeded	Administrative	Informational	2024-05-07T10:...	387407e5-94af...	Azure Smart Al...	microsoft.ir
41ba2496-1e58...	Update web sit...	Started	Administrative	Informational	2024-05-07T10:...	387407e5-94af...	cloudlearning4...	microsoft.w
a65c54d7-d9ce...	Create Deploy...	Started	Administrative	Informational	2024-05-07T07:...	387407e5-94af...	cloudlearning4...	Microsoft.R
a03fb719-45d3...	Update hosting...	Started	Administrative	Informational	2024-05-06T10:...	387407e5-94af...	cloudlearning4...	Microsoft.V
a3063ea6-945f...	Validate Deploy...	Succeeded	Administrative	Informational	2024-05-06T08:...	387407e5-94af...	cloudlearning4...	Microsoft.R

00:00:04 Query executed successfully.