**ADF: Azure DataLake Connector – Move data to and from Azure DataLake**

The document outlines how you can use the Copy Activity in Azure Data Factory to move data from another data store to Azure Data Lake and move data from Azure Data Lake to another data store.

# Sample: Copy Data from Azure Blob to ADL

The sample includes the following:

1. A linked service of type ‘AzureStorage’
2. A linked service of type ‘AzureDataLake’
3. An input dataSet of type ‘AzureBlob’
4. An output dataSet of type ‘AzureDataLake’
5. A pipeline with a Copy activity that uses ‘BlobSource’ and ‘AzureDataLakeSink’

The sample copies data belonging to a time series from an Azure Blob to Azure Data Lake every hour. The JSON properties used in these samples are described in sections following the samples.

**Azure Storage linked service**:

{

"name": "StorageLinkedService",

"properties": {

"type": "AzureStorage",

"typeProperties": {

"connectionString": "DefaultEndpointsProtocol=https;AccountName=<accountname>;AccountKey=<accountkey>"

}

}

}

**Azure DataLake linked service**:

{

"name": "AzureDataLakeLinkedService",

"properties": {

"type": "AzureDataLake",

"typeProperties": {

"dataLakeUri": "https://<accountname>.azuredatalake.net/webhdfs/v1",

"authorization": "<authorization URL>"

}

}

}

**Azure Blob Input dataset**:

Data is picked up from a new blob every hour (frequency: Hour, interval: 1). The folder path and file name for the blob are dynamically evaluated based on the start time of the slice that is being processed. The folder path uses year, month, and day part of the start time and file name uses the hour part of the start time. “External”: “true” setting informs the Data Factory service that this table is external to the data factory and not produced by an activity in the data factory.

{

"name": "AzureBlobInput",

"properties": {

"type": "AzureBlob",

"linkedServiceName": "StorageLinkedService",

"typeProperties": {

"Path": "mycontainer/myfolder/yearno={Year}/monthno={Month}/dayno={Day}",

“partitionedBy": [

{

"name": "Year",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "yyyy"

}

},

{

"name": "Month",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%M"

}

},

{

"name": "Day",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%d"

}

},

{

"name": "Hour",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%H"

}

}

]

},

"external": true,

"availability": {

"frequency": "Hour",

"interval": 1

},

"policy": {

"externalData": {

"retryInterval": "00:01:00",

"retryTimeout": "00:10:00",

"maximumRetry": 3

}

}

}

}

**Azure DataLake output dataset**:

The sample copies data to Azure DataLake. New data is copies to DataLake every hour.

{

"name": "AzureDataLakeOutput",

"properties": {

"type": "AzureDataLake",

"linkedServiceName": " AzureDataLakeLinkedService",

"typeProperties": {

"Path": "mycontainer/myfolder/yearno={Year}/monthno={Month}/dayno={Day}",

"partitionedBy": [

{

"name": "Year",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "yyyy"

}

},

{

"name": "Month",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%M"

}

},

{

"name": "Day",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%d"

}

},

{

"name": "Hour",

"value": {

"type": "DateTime",

"date": "SliceStart",

"format": "%H"

}

}

]

},

"external": true,

"availability": {

"frequency": "Hour",

"interval": 1

},

"policy": {

"externalData": {

"retryInterval": "00:01:00",

"retryTimeout": "00:10:00",

"maximumRetry": 3

}

}

}

}

**Pipeline with a Copy Activity**:

The pipeline contains a Copy Activity that is configured to use the above input and output datasets and is scheduled to run every hour. In the pipeline JSON definition, the **source** type is set to **BlobSource** and **sink** type is set to **AzureDataLakeSink**.

{

"name":"SamplePipeline",

"properties":{

"start":"2014-06-01T18:00:00",

"end":"2014-06-01T19:00:00",

"description":"pipeline with copy activity",

"activities":[

{

"name": "AzureBlobtoDataLake",

"description": "Copy Activity",

"type": "Copy",

"inputs": [

{

"name": "AzureBlobInput"

}

],

"outputs": [

{

"name": "AzureDataLakeOutput"

}

],

"typeProperties": {

"source": {

"type": "BlobSource",

"treatEmptyAsNull": true,

"blobColumnSeparators": ","

},

"sink": {

"type": "AzureDataLakeSink"

}

},

"scheduler": {

"frequency": "Hour",

"interval": 1

},

"policy": {

"concurrency": 1,

"executionPriorityOrder": "OldestFirst",

"retry": 0,

"timeout": "01:00:00"

}

}

]

}

}

## Azure Storage Linked Service properties

You can link an Azure storage account to an Azure data factory using an Azure Storage linked service. The following table provides description for JSON elements specific to Azure Storage linked service.

| PROPERTY | DESCRIPTION | REQUIRED |
| --- | --- | --- |
| type | The type property must be set to: **AzureStorage** | Yes |
| connectionString | Specify information needed to connect to Azure storage for the connectionString property. You can get the connectionString for the Azure storage from the Azure Portal. | Yes |

## Azure DataLake Linked Service properties

The following table provides description for JSON elements specific to Azure DataLake linked service.

| PROPERTY | DESCRIPTION | REQUIRED |
| --- | --- | --- |
| type | The type property must be set to: **AzureDataLake** | Yes |
| dataLakeUri | Specify information about the Azure DataLake account. It is in the following format:  https://<accountname>.azuredatalake.net/webhdfs/v1  <accountname> = Azure DataLake account name | Yes |
| authorization | In order to get the authorization URL above, you need to do the following:   1. Login using your org/live id to the following url:   <https://login.microsoftonline.com/common/oauth2/authorize?response_type=code&client_id=a306baf0-5ad8-4f6f-babf-6a286b0142ba&redirect_uri=https://datafactory.azure.com/oauthredirect>   1. After logging in, copy the URL from the browser to the ‘Azure DataLake linked service’ above in the “<authorization URL>”. 2. Then deploy the above linked service.   **Note**: This is a temporary manual step. This will go away once we complete the ADF OAuth UI work. |  |

## Azure Blob type properties

For a full list of JSON sections & properties available for defining datasets, see the [Creating datasets](https://azure.microsoft.com/en-us/documentation/articles/data-factory-create-datasets/) article. Sections like structure, availability, and policy of a dataset JSON are similar for all dataset types (Azure SQL, Azure blob, Azure table, etc...).

The **typeProperties** section is different for each type of dataset and provides information about the location, format etc. of the data in the data store. The typeProperties section for dataset of type **AzureBlob** dataset has the following properties.

| PROPERTY | DESCRIPTION | REQUIRED |
| --- | --- | --- |
| folderPath | Path to the container and folder in the blob storage. Example: myblobcontainer\myblobfolder | Yes |
| fileName | Name of the blob. fileName is optional.  If you specify a filename, the activity (including Copy) works on the specific Blob.  When fileName is not specified Copy will include all Blobs in the folderPath for input dataset.  When fileName is not specified for an output dataset, the name of the generated file would be in the following this format: Data..txt (for example: : Data.0a405f8a-93ff-4c6f-b3be-f69616f1df7a.txt | No |
| partitionedBy | partitionedBy is an optional property. You can use it to specify a dynamic folderPath and filename for time series data. For example, folderPath can be parameterized for every hour of data. See the Leverage partitionedBy prperty section below for details and examples. | No |
| format | Two formats types are supported: **TextFormat**, **AvroFormat**. You need to set the type property under format to either of these values. When the format is TextFormat you can specify additional optional properties for format. See the [Specifying TextFormat](https://azure.microsoft.com/en-us/documentation/articles/data-factory-azure-blob-connector/#specifying-textformat) section below for more details. | No |

## Azure DataLake Dataset type properties

For a full list of JSON sections & properties available for defining datasets, see the [Creating datasets](https://azure.microsoft.com/en-us/documentation/articles/data-factory-create-datasets/) article. Sections like structure, availability, and policy of a dataset JSON are similar for all dataset types (Azure SQL, Azure blob, Azure table, etc...).

The **typeProperties** section is different for each type of dataset and provides information about the location, format etc. of the data in the data store. The typeProperties section for dataset of type **AzureBlob** dataset has the following properties.

| PROPERTY | DESCRIPTION | REQUIRED |
| --- | --- | --- |
| Path | Path to the container and folder in the azure data lake. | Yes |
| partitionedBy | partitionedBy is an optional property. You can use it to specify a dynamic folderPath for time series data. For example, folderPath can be parameterized for every hour of data. See the Leverage partitionedBy prperty section below for details and examples. | No |

### **Leveraging partitionedBy property**

As mentioned above, you can specify a dynamic folderPath and filename for time series data with the **partitionedBy** section, Data Factory macros and the system variables: SliceStart and SliceEnd, which indicate start and end times for a given data slice.

See [Creating Datasets](https://azure.microsoft.com/en-us/documentation/articles/data-factory-create-datasets/) and [Scheduling & Execution](https://azure.microsoft.com/en-us/documentation/articles/data-factory-scheduling-and-execution/) articles to understand more details on time series datasets, scheduling and slices.

#### **Sample 1**

Copy

"folderPath": "wikidatagateway/wikisampledataout/{Slice}",

"partitionedBy":

[

{ "name": "Slice", "value": { "type": "DateTime", "date": "SliceStart", "format": "yyyyMMddHH" } },

],

In the above example {Slice} is replaced with the value of Data Factory system variable SliceStart in the format (YYYYMMDDHH) specified. The SliceStart refers to start time of the slice. The folderPath is different for each slice. For example: wikidatagateway/wikisampledataout/2014100103 or wikidatagateway/wikisampledataout/2014100104

#### **Sample 2**

Copy

"folderPath": "wikidatagateway/wikisampledataout/{Year}/{Month}/{Day}",

"fileName": "{Hour}.csv",

"partitionedBy":

[

{ "name": "Year", "value": { "type": "DateTime", "date": "SliceStart", "format": "yyyy" } },

{ "name": "Month", "value": { "type": "DateTime", "date": "SliceStart", "format": "MM" } },

{ "name": "Day", "value": { "type": "DateTime", "date": "SliceStart", "format": "dd" } },

{ "name": "Hour", "value": { "type": "DateTime", "date": "SliceStart", "format": "hh" } }

],

In the above example, year, month, day, and time of SliceStart are extracted into separate variables that are used by folderPath and fileName properties.

## Azure Blob Copy Activity type properties

For a full list of sections & properties available for defining activities, see the [Creating Pipelines](https://azure.microsoft.com/en-us/documentation/articles/data-factory-create-pipelines/) article. Properties like name, description, input and output tables, various policies etc are available for all types of activities.

Properties available in the typeProperties section of the activity on the other hand vary with each activity type and in case of Copy activity they vary depending on the types of sources and sinks

**BlobSource** supports the following properties in the **typeProperties** section:

| PROPERTY | DESCRIPTION | ALLOWED VALUES | REQUIRED |
| --- | --- | --- | --- |
| treatEmptyAsNull | Specifies whether to treat null or empty string as null value. | TRUE FALSE | No |
| skipHeaderLineCount | Indicate how many lines need be skipped. It is applicable only when input dataset is using**TextFormat**. | Integer from 0 to Max. | No |

**BlobSink** supports the following properties **typeProperties** section:

| PROPERTY | DESCRIPTION | ALLOWED VALUES | REQUIRED |
| --- | --- | --- | --- |
| blobWriterAddHeader | Specifies whether to add header of column definitions. | TRUE FALSE (default) | No |

## Specifying structure definition for rectangular datasets

The structure section in the datasets JSON is an **optional** section for rectangular tables (with rows & columns) and contains a collection of columns for the table. You will use the structure section for either providing type information for type conversions or doing column mappings. The following sections describe these features in detail.

Each column contains the following properties:

| PROPERTY | DESCRIPTION | REQUIRED |
| --- | --- | --- |
| name | Name of the column. | Yes |
| type | Data type of the column. See type conversions section below for more details regarding when should you specify type information | No |
| culture | .NET based culture to be used when type is specified and is .NET type Datetime or Datetimeoffset. Default is “en-us”. | No |
| format | Format string to be used when type is specified and is .NET type Datetime or Datetimeoffset. | No |

The following sample shows the structure section JSON for a table that has three columns userid, name, and lastlogindate.

Copy

"structure":

[

{ "name": "userid"},

{ "name": "name"},

{ "name": "lastlogindate"}

],

Please use the following guidelines for when to include “structure” information and what to include in the **structure** section.

1. **For structured data sources** that store data schema and type information along with the data itself (sources like SQL Server, Oracle, Azure table etc.), you should specify the “structure” section only if you want do column mapping of specific source columns to specific columns in sink and their names are not the same (see details in column mapping section below).

As mentioned above, the type information is optional in “structure” section. For structured sources, type information is already available as part of dataset definition in the data store, so you should not include type information when you do include the “structure” section.

1. **For schema on read data sources (specifically Azure blob)** you can chose to store data without storing any schema or type information with the data. For these types of data sources you should include “structure” in the following 2 cases:
   1. You want to do column mapping.
   2. When the dataset is a source in a Copy activity, you can provide type information in “structure” and data factory will use this type information for conversion to native types for the sink. See [Move data to and from Azure Blob](https://azure.microsoft.com/en-us/documentation/articles/data-factory-azure-blob-connector/) article for more information.

### **Supported .NET-based types**

Data factory supports the following CLS compliant .NET based type values for providing type information in “structure” for schema on read data sources like Azure blob.

* Int16
* Int32
* Int64
* Single
* Double
* Decimal
* Byte[]
* Bool
* String
* Guid
* Datetime
* Datetimeoffset
* Timespan

For Datetime & Datetimeoffset you can also optionally specify “culture” & “format” string to facilitate parsing of your custom Datetime string. See sample for type conversion below.

### **Type conversion sample**

The following sample is for copying data from a Blob to Azure SQL with type conversions.

Suppose the Blob dataset is in CSV format and contains 3 columns. One of them is a datetime column with a custom datetime format using abbreviated French names for day of the week.

You will define the Blob Source dataset as follows along with type definitions for the columns.

Copy

{

"name": "AzureBlobTypeSystemInput",

"properties":

{

"structure":

[

{ "name": "userid", "type": "Int64"},

{ "name": "name", "type": "String"},

{ "name": "lastlogindate", "type": "Datetime", "culture": "fr-fr", "format": "ddd-MM-YYYY"}

],

"type": "AzureBlob",

"linkedServiceName": "StorageLinkedService",

"typeProperties": {

"folderPath": "mycontainer/myfolder",

"fileName":"myfile.csv",

"format":

{

"type": "TextFormat",

"columnDelimiter": ","

}

},

"external": true,

"availability":

{

"frequency": "Hour",

"interval": 1

},

"policy": {

"externalData": {

"retryInterval": "00:01:00",

"retryTimeout": "00:10:00",

"maximumRetry": 3

}

}

}

}