# AZURE DATA FACTORY-PASSING PARAMETERS

This articles walks you through how to pass parameters between a pipeline and activity as well as between the activities

#### Written By-

Blesson John (Data Solution Architect-Microsoft)Issagha BA (Data Solution Architect-Microsoft)

#### **Reviewed By-**

Ye Xu (Senior Program Manager-ADF) Gaurav Malhotra (Principal Program Manager) Passing parameter between pipeline and activity as well as between activities

# Contents

Azure data Factory – Passing Parameters	2
Passing parameter via pipeline	2
Passing parameter between activities	11
The architecture	11
The step by step process	11

© 2019 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners

# Azure data Factory – Passing Parameters

Passing parameters to ADF is quite important as it provides the flexibility required to create dynamic pipelines. To reference a parameter, one will have to provide the fully qualified name of the parameter. It is worth noting that parameter names are case sensitive. A parameter could be a user input, which means that the parameter is passed from the pipeline layer or could be an input coming from an activity within the pipeline.

### Passing parameter via pipeline

In this example, I am creating a pipeline that will use dynamic REST API URL to extract data in JSON format and move it to a blob store. Click "Author & Monitor" to create a new pipeline. A new tab will appear, and it will look like the one below-



The base URL we will be using is <a href="https://conferenceapi.azurewebsites.net/">https://conferenceapi.azurewebsites.net/</a>

Select "Create pipeline" icon. Rename the pipeline by replacing "pipeline1" with name of your choice. It is worth following some naming convention.

Activities * « Search Activities Move & Transform Batch Service Data Lake Analytics General HDInsight I teration & Conditionals Machine Learning + - A E & S & X & - = = General Parameters Variables Output Name * Description Concurrency Latent intervence Output Name * Description Concurrency Latent intervence Output Name * Description Concurrency Latent intervence Output Name * Description Concurrency Latent intervence Output		
> Search Activities         > Move & Transform         > Batch Service         > Databricks         > Data Lake Analytics         > General         > HDInsight         > Iteration & Conditionals         > Machine Learning         + - A	Activities × «	🚰 Save as template 🗸 Validate 🕨 Debug 🚱 Add trigger
Move & Transform       Batch Service       Databricks       Data Lake Analytics       General       HDInsight       Iteration & Conditionals       Machine Learning         + - A         General         Parameters       Variables       Output         Name *       pipeline1          Description       Concurrency	♀ Search Activities	
Batch Service       Databricks       Data Lake Analytics       General       HDInsight       Iteration & Conditionals       Machine Learning         + - A         General         Parameters       Variables         Output         Name *       pipeline1         Description         Concurrency	Move & Transform	
<ul> <li>Databricks</li> <li>Data Lake Analytics</li> <li>General</li> <li>HDInsight</li> <li>Iteration &amp; Conditionals</li> <li>Machine Learning</li> <li>+ - A E &amp; C IN Pa - 5</li> <li>General Parameters Variables Output</li> <li>Name * pipeline1</li> <li>Description</li> <li>Concurrency</li> </ul>	Batch Service	
<ul> <li>Data Lake Analytics</li> <li>General</li> <li>HDInsight</li> <li>Iteration &amp; Conditionals</li> <li>Machine Learning</li> <li>+ - A E &amp; C IN Pa</li> <li>General Parameters Variables Output</li> <li>Name * pipeline1</li> <li>Description</li> <li>Concurrency</li> </ul>	Databricks	
Ceneral HDInsight Iteration & Conditionals Machine Learning	Data Lake Analytics	
<ul> <li>HDInsight</li> <li>Iteration &amp; Conditionals</li> <li>Machine Learning</li> <li>+ - A &amp; A</li></ul>	General	
Iteration & Conditionals     Machine Learning     + - A & A R R R R     General Parameters Variables Output     Name *      pipeline1     Description     Concurrency	HDInsight	
Machine Learning       +     -     -     -       General     Parameters     Variables     Output       Name *     pipeline1       Description	Iteration & Conditionals	
+ - A & S S S	Mashina Learning	
+ - A 6 C IX 12	<ul> <li>Machine Learning</li> </ul>	
+     -     A     B     C <th></th> <th></th>		
General     Parameters     Variables     Output       Name *     ippeline1       Description       Concurrency		+ - A 🖻 X 🕅 🖬 📲
Name * pipeline 1 Description Concurrency		General Parameters Variables Output
Name * pipeline1 Description Concurrency		
Description Concurrency		Name *
Concurrency •		
Concurrency		Description
		Description
		Description Concurrency

Example of naming convention-

ADF Component	Naming Convention	Example
pipeline	pl_businessfunction_nnnn	pl_financereporting_0001*
dataset	ds_technologyname_nnnn	ds_sql_finance_0001*
activity	ac_techfunction_nnnn	ac_copy_blob_stg_sales_0001*
linked service	ls_connectiontype_nnnn	ls_sql_oreaserver_0001*

\*we are using the numbering to get around limits such as maximum number of activities in a pipeline.

Δftor	renaming	the r	nineline	select the	narameter t	ah
Aiter	renaming	uie p	npenne,	select the	parameter t	.au

General	Parameters	Variables	Output		
Name *	pl_	copydata			
Description					
				1	
Concurrency				0	
Annotations	+	- New			

Once parameter tab is clicked, you can hit "+New" icon. Type in the name-**relativeurl**-with type as **string** and default value as **/speakers**.



Either use the search bar to find copy data activity or expand "Move & Transform" node to find copy data activity. Drag and drop this activity to the white canvas on your right

A	ctivities	≽	~	
,	Search Activities			
4	Move & Transfor	m		
•	Copy Data			
	Data Flow (Previe	w)		
۲	Batch Service			
۲	Databricks			
۲	Data Lake Analy	tics		
۲	General			
۲	HDInsight			
Þ	Iteration & Cond	litior	nals	
۲	Machine Learnin	g		-

Once copied to the right pane, your screen will appear like the one below-

III pl-copydata * $ imes$								
Activities ×	× «	🛃 Save as templ	ate	$\checkmark$	Validate	Deb	bug	Q⊕ Add trigger
♀ Search Activities						$\bigcirc$		
Move & Transform	ı		Copy D	ata				
🖥 Copy Data			r,	Сор	y Data1			
Data Flow (Preview)			Ē	0	$\Box$	( <b>+</b> )		
Batch Service								
Databricks								
Data Lake Analytic	cs							
General								

Within the copy activity, we need to define the source and sink. In this case, the source will be the REST API and the sink will be a blob store.

#### **Defining Source**

Click the **Source** tab and click the icon "**+New**". When the list of new datasets appears, search for "Rest" and select "Rest".



# Set the name of the dataset in the general tab.

General	Connection <sup>1</sup> Parameters	
Name *	ds_restapi	
Description		
Annotations	+ New	

General	Connectior	n <sup>1</sup>	Parameters				
Linked service *		Select		•	+ N	lew	
Relative Url	[				69	Preview data	
Request Method		GET		-			

# Select the Connection tab, and create the linked service by clicking the "+New" icon

# Fill the information as given below and hit the test connection button-

New Linked Service (F	New Linked Service (REST)					
Name *						
ls_restapi						
Description						
Connect via integration runtime *		0				
AutoResolveIntegrationRuntime		•				
Base URL *						
https://conferenceapi.azurewebsites.net/						
Authentication type *						
Anonymous		-				
Server Certificate Validation		0				
Enable	Disable					
Annotations						
+ New						
<ul> <li>Advanced 0</li> </ul>						





# Click finish and select the parameters tab

General	Connection	Parameters	
+ New	Delete		
NAME		ТҮРЕ	DEFAULT VALUE
relative	eurl	String •	/speaker

#### Now let us parameterize the Relative Url

General	Connection Parameters			
Linked service *	🔇 ls_restapi	✓ Test connection	🖉 Edit 🛛 🕂 I	New
Relative Url	Add dynamic content [Alt+P]	<del>රට</del> Preview data		
Request Method	GET	•		
<ul> <li>Additional H</li> </ul>	eaders			

Any field that comes up with the "Add dynamic content" allows parameterization. We need to set the Relative Url to parameter that is coming from pipeline. Earlier, I stated that we need to use the fully qualified name. Double click "relativeurl" and click finish

@dataset().relativeurl	
Clear Contents	
,	+
Use <u>expressions, functions</u> or refer to <u>system variables</u> .	
<ul> <li>Functions</li> </ul>	
Sex Expand All	
Collection Functions	
<ul> <li>Conversion Functions</li> </ul>	
Conversion Functions     Date Functions	
Conversion Functions     Date Functions     Logical Functions	
<ul> <li>Conversion Functions</li> <li>Date Functions</li> <li>Logical Functions</li> <li>Math Functions</li> </ul>	
<ul> <li>Conversion Functions</li> <li>Date Functions</li> <li>Logical Functions</li> <li>Math Functions</li> <li>String Functions</li> </ul>	

At the source page, set the dataset level parameter to the pipeline level parameter. **You cannot see the pipeline level parameter at the dataset level.** This is a very important thing to note.

General	Source	Sink <sup>1</sup>	Mapping	Settings	User Pro	perties	
Source datase	et *	🤣 RestReso	urce1	•	🖉 Edit	+ New	60 Preview data
		<ul> <li>Dataset pro</li> </ul>	perties 0				
		NAME	VALUE	E			
		relativeurl	@p	peline().paramet	ers.relativeurl		

#### **Defining Sink**

Click the **Sink** tab and click the icon "+New".

General	Source	Sink <sup>1</sup>	Mapping	Settings	User Properties
Sink dataset *		Select		▼	+ New

When the list of new datasets appears, search for "Azure data lake storage gen2" and select "Azure data lake storage gen2".



#### Select the format to be Json



In the general tab, fill the details using the naming convention that is used by your team

General	Connection <sup>1</sup>	Schema	Parameters	
Name *	ds_dat	alakegen2		]
Description				
Annotations	+	New		al d

Click on the Connection tab and setup the linked service. Use managed identity where possible.

Name *	
ls_azuredatalakegen2	
Description	
Connect via integration runtime *	
AutoResolveIntegrationRuntime	•
Authentication method	
Managed Identity	•
Account selection method From Azure subscription	Enter manually
Azure subscription	0
Select all	•
Storage account name *	
comfydatalake	•
Managed identity application ID: 5794d07d-8c Grant data factory managed identity access to Annotations + New	25-4eab-b318-a491a8313eaf your Azure Data Lake Storage Gen2. Details
Managed identity application ID: 5794d07d-8c Grant data factory managed identity access to Annotations + New • Advanced •	23-factub-b116-461184311Berl your Azum Data Lake Storage Gen2. <b>Details</b>
Managed identity application ID: 5794/d076-86 Grant data factory managed identity access to Annotations + New Advanced 0	25-desb-b316-d913d313eaf your Azure Data Lake Storage Gen2. <b>Detail:</b>
Managad dentity application ID: 5784/07/6-86 Grant data factory managed identity access to Annotations + New Advanced ©	23-6asb-b316-4911481116arl your Azure Data Lake Storage Gen2. <b>Details</b>
Atranged Learny application D. 37944076 de Grant data factory managed identity access to Annotations + New Advanced •	23-6asb-b316-49114311teaf your Azure Data Lake Storage Gen2. <b>Details</b>
Menagai dentity replication (D. 3744076 etc Grant data factiony managed identity access to Annotations + New > Advanced •	25-6asb-b316-a611a8315asf your Axure Data Lake Storage Gen2. <b>Details</b>
Managad elevity replacation 0: J7946076 & Grant data factory managed identify access to Annotations + New > Advanced ©	25-6asb-b316-4011481116arl your Azure Data Lake Storage Gen2. <b>Detait</b> :

Once the destination has been setup, the connection page will look like the one below-

General Connec	tion Schema Paramet	ers		
Linked service *	Is_azuredatalakegen2			+ New
File path	blobcopy	/ File	0	Browse
Compression type	None	•		
Filter by last modified	Start time (UTC)	End time (UTC)	0	
Binary copy	0			
<ul> <li>File format settings</li> </ul>				
File format	JSON format			
File pattern	Set of objects	•		
<ul> <li>JSON path settings</li> </ul>				
Cross-apply nested JSON array	None	Parse JSON Path		

To save this, we need to click the publish all icon at the top left-

L	🖞 Publish All 2 🗸	Validate	All 📿 Refresh 🗎	Discard All	Data Flow Debug	🕼 ARM Template 🗡	⊳ «
<	🗊 pl_copydata * 🗙	⊞ ds_r	estapi * ×				
-	Activities ×	: «	Save as template	🗸 Validate	Debug	🚱 Add trigger	🚺 Code

Once the pipeline has been published successfully, you can trigger the pipeline to test it. Select trigger now to trigger the pipeline.



Click on the monitor icon on the left and check whether the pipeline has completed successfully

»	>>      Ø Dashboards III Pipeline Runs III Tigger Runs III III Alerts & Metrics												
	-	∂ Run (	Cancel option	ns 📿 Refresh									
<b>%</b>	]	Last 24 Hours	05/14/2019 9:47	AM - 05/15/2019 9:47	AM V	Time Zone	(UTC+00:00) Dub	olin, Edinburgh, Li 🔹	View All Reru	in History			<b>V</b> eFilter
		Bingling N	In Progress	Actions	Cancelled	*	Duration	Triggered By	Status	Paramotors	Annotations V	Error	PupID
	Ľ	pl_copydat	anne u	The second secon	05/15/2019	9, 3:45:33 PM	00:00:08	Manual trigger	Succeeded	(@)	Annotations	Enor	78f6c3ee-351c-41b8-

# Passing parameter between activities

Before we even kick off this session, it is important to understand what an activity is. Activity is where a particular action is performed. The action could be to cleanse data or update a control table within SQL Database. An activity is within a pipeline, where the pipeline is a logical container having one or more activities.

We will be creating a pipeline that will copy the meta data information of a file in blob store and then loads this information into an Azure SQL database table. All the information will be moving between activities via parameters-the output parameters.

### The architecture

We will follow a three-step process to show how to pass parameters between activities. The flat file currently exists on ADLS gen 1 storage. We use ADF activity to get the meta data about a file stored in ADLS gen 1. Once this information is available, we use parameters to pass the details to the next activity, which is a stored procedure within Azure SQL database.



#### The step by step process

Go to a data factory and create a new pipeline and drag the activity "Get Metadata".

👌 Azure DevOps GIT 🗸 🦯 👂	DEV branch 🖂 🔚 Save All	🖞 Publish 🛛 🗸 Valid	ate All 🗧 Refresh 🔋 Discard All	💽 Data Flow Debug 🛛 🕺 ARM Template 🗸	⊳ «
Factory Resources 👻 «	03 Copture_File_In • ×				
,9 Filter resources by name +	Activities * «	🗟 Save 🔣 Save ar	semplate 🗸 Validate 🕨 Debug	😪 Add tripper	🔝 Code
4 Pipelines 2	$\mathcal{P}$ meta		0		
CopyFinanceData_001	▲ General		Get Metadata		
03 * Capture, File, Information	<ul> <li>Get Metadata</li> </ul>		(i) GetFileEditInfo		
Datasets 2					
<ul> <li>Data Flows (Preview)</li> </ul>					
<ul> <li>Templates</li> </ul>					
		+ - & M General Dataset	St. [3] 33 •8 User Properties	_	
			(	7	
		Name *	Gaterias and		
		Description			
		Timeout	7.00.00.00		
		Retry	0	0	
		Retry interval	30	0	
X Connections		Secure output	0		
(b) Triacert		Common Security			

We plan to find the last modified date of a file in ADLS storage. Set the dataset to be the file stored in the ADLS storage. I am planning to capture three arguments- Item Name, Item Type and Last Modified.

Activities × «	🔚 Save	🛃 Save as	template	🗸 Validate	Debug	🚱 Add trigger	
∽ meta							
▲ General		ot Motodata		0			
🛈 Get Metadata		j GetFileE	EditInfo ì ⊕				
	+ -	603	8 18	□↑ ■■ ↓□			
	+ — General	A [er] Dataset	S. IN User F	<b>₽</b> roperties		_	
	+ General Dataset * Field list	A [07] Dataset	S. S. User f Finance + New	Properties resource v I Dele	• Ite	— D Edit	+ New
	+ - General Dataset * Field list	A [w] Dataset	N   N     User F     Image: second se	T         T         eroperties         esource         v	• te	Edit	+ New
	+ General Dataset * Field list	A [293] Dataset	State   IN     User f     Image: state	The     ■       Properties       ceSource       v     I       I     Dele       MENT       Name	▼ te	 ℓ∂ Edit	+ New

To pass these arguments to the next activity, we need to first debug the pipeline. You will be able to see the outcome in the output tab.

General	Parameters	Variables	Output				
Pipeline Run ID	: b9a75b90-cedb-	4463-95f5-e30a6	14696a6	[@]	$\langle \rangle$	0	

Once the execution completes, you have two option within action. One is the input and the other is output.

NAME	ТҮРЕ	RUN START	DURATION	STATUS	ACTIONS	RUNID
GetFileEditInfo	GetMetadata	05/31/2019 2:48 PM	00:00:11	Succeeded	<u>→</u> →	57cbbcf5-ea99-49fd-aab9-65cf8e5a1b93

In this case, click on the  $\ominus$  symbol and view whether the three arguments we selected are displayed.

	Output		ZX	
out	{ " <mark>itemName"</mark> : "Address "itemType": "File", " <mark>lastModified</mark> ": "2019- "effectiveIntegrationR (North Europe)", "executionDuration":	s", -03-27T15:36:14Z", untime": "DefaultIntegi 15	rationRuntime	
a6 🕼	}			
00:00:11	Succeeded	→ →	57cbbcf5-ea99-49fc	d-aab9

Create a table and stored procedure within the Azure database that was created. The code is provided below.

```
-- Create Stored Procedure Template for Azure SQL Database
--________
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
- -
_____
==
-- Author:
           John Doe
-- Create Date: 31/05/2019
-- Description: This stored procedure will populate the data in the control table
- -
_____
===
CREATE PROCEDURE usp_populate_control_table
(
   -- Add the parameters for the stored procedure here
   @V_Item_Name varchar(255),
     @V_Item_Type varchar(255),
     @D_Last_Modified datetime
)
AS
BEGIN
     SET NOCOUNT ON
     BEGIN TRY
          INSERT INTO TB FILE METADATA
          (
           V ITEM NAME,
           V_ITEM_TYPE,
           D_LAST_MODIFIED
          )
          VALUES(@V_Item_Name,
                 @V_Item_Type,
                 @D_Last_Modified
          )
     END TRY
     BEGIN CATCH
          SELECT
               ERROR_NUMBER() AS ErrorNumber
            ,ERROR_MESSAGE() AS ErrorMessage;
     END CATCH
END
GO
```

Drag the stored procedure activity into the ADF canvas.

💷 Capture_File_In * 🗙	⊞	FinanceSource	×		
Activities × «	×	🔚 Save	🚰 Save as template	🗸 Validate	Debug 🚱 Add trigger
angle stored					0
General					Stored Procedure
Stored Procedure			Get Metadata		Stored Procedure1

Now connect the activities such that when "get metadata" activity completes successfully, you will execute the stored procedure. Just as in SSIS, you have option such as on failure, completion, success etc.



Setup the SQL account as shown below

🟥 Publish 🗸 Validate All 📿 Refresh 🗌		
FinanceSource ×	Name *	
🖓 Sava 💭 Sava at templata 💦 Malidate	. and and management	
	Description	
Get Metadata	Type *	
	Azure SQL Database	•
GetFileEditInfo	Connect via integration runtime *	0
	AutoResolveIntegrationRuntime	•
	Connection String	Azure Key Vault
	Account selection method From Azure subscription	• Enter manually
	Fully qualified domain name *	
	database.windows.net	
+ - A 🖻 🕅 🏗 📲	Database name *	
General SQL Account 1 Stored Procee	u	
	Authentication type *	
Linked service * Select	SQL Automation	
	User name *	
	Password	Azure Key Vault
	Password *	
	Additional connection properties	
		Connection successful
	Cancel	Test connection Finish

Select the stored procedure that was created "usp\_populate\_control\_table"

General	SQL Accou	unt Stored Procedure	User Properties	
▲ Details				
Stored procedure	e name *	[dbo].[usp_populate_control_table	▼ C Refresh	

You can either import the parameters or manually enter the parameter. When you use import parameters, the input parameters for the stored procedure are identified automatically.

Import parameter			
Stored procedure paramet	ers <sup>0</sup>		
NAME	ТҮРЕ	VALUE	
D_Last_Modified	DateTime	Value	Treat as null
V_Item_Name	String	Value	Treat as null
V_Item_Type	String	Value	Treat as null

Click on the Value text box and press "add dynamic content"

١	1	Δ	I	ι	J	F
					-	_

Value	Treat as null
Add dynamic content [Alt+P]	

 $\times$ 

Select the Activity output or you can manually type the value

# Add Dynamic Content

@activity('GetFileEditInfo').output
Clear Contents
Use <u>expressions, functions</u> or refer to <u>system variables</u> .
ID of the specific pipeline run
Pipeline trigger ID ID of the trigger that invokes the pipeline
Pipeline trigger name Name of the trigger that invokes the pipeline
Pipeline trigger time Time when the trigger that invoked the pipeline. The trigger time is the actual fired time, not the sch
Pipeline trigger type Type of the trigger that invoked the pipeline (Manual, Scheduler)
Functions
✓ Expand All
Collection Functions
Conversion Functions
Date Functions
Logical Functions
Math Functions
String Functions
Activity outputs
GetFileEditInfo GetFileEditInfo activity output
Cancel

The first selection will look something like the one below.



The names are case sensitive and once completed the screen will show up as follows

⊟ <mark>select *</mark> from TB_F	NAME	TYPE DateTime A	VALUE	f(e'),output.lastModified
■select * from TB_F	D.Last.Modified	DateTime	@activity['GetFileEdttr	fo').output.lastModified
■select * from TB_F	FILE_METADAT	A		
From TB_F	FILE_METADAT.	A		
[from TB_F	FILE_METADAT.	A		
0.% - 1				
5 78 - 1				
🗄 Results 📑	Messages			
I_ID V_IT				
	TEM NAME V	ITEM TYPE	D LAST MODIFIED	
1 i 1 i Add	TEM_NAME V	_ITEM_TYPE	D_LAST_MODIFIED 2019-03-27 15:36:14 00	D

You could now debug the pipeline and if everything looks good you will see that the table in azure SQL database is populated.

This indicates that the file was last modified in March 2019.