Variables in Azure Data Factory

This post is part 21 of 25 in the series Beginner's Guide to Azure Data Factory

In the previous post, we talked about why you would want to build a dynamic solution, then looked at how to use parameters. In this post, we will look at variables, how they are different from parameters, and how to use the set variable and append variable activities.

Variables

Parameters are external values passed *into* pipelines. They can't be changed *inside* a pipeline. **Variables**, on the other hand, are internal values that *live inside* a pipeline. They *can* be changed inside that pipeline.

Parameters and variables can be completely separate, or they can work together. For example, you can pass a parameter *into* a pipeline, and then use that parameter value in a **set variable** or **append variable** activity.

System Variables

In the previous post, I called out the syntax for dataset and pipeline parameters. Did you notice the difference? :)

```
@dataset().ParameterName
@pipeline().parameters.ParameterName
```

In a dataset, you can reference the parameter name directly. But in a pipeline, you have to first reference "parameters". That's because inside a pipeline, you have both parameters *and* **system variables**:

To use **system variables**, you reference them in a similar way to parameters:

```
@pipeline().DataFactory
@pipeline().Pipeline
@pipeline().RunId
@pipeline().TriggerId
@pipeline().TriggerName
@pipeline().TriggerTime
@pipeline().TriggerType
```

User Variables

User variables are slightly different, though. You reference them by their name:
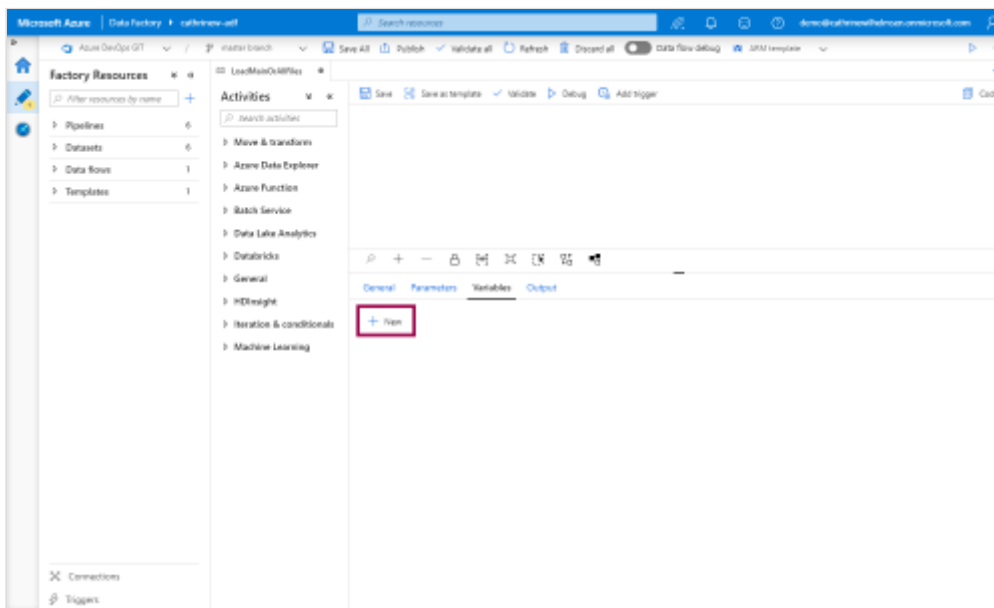
```
@variables('VariableName')
```

Let's see how this works!

In this example, we will build a new pipeline to show how a variable can be used and updated. Our goal is to create a pipeline that can be used to load *some* or *all* the files from Rebrickable.

Create Variables

Create a new pipeline, go to the variables properties, and click **+ new**:

Give the variable a **name** and choose the **type**. You can specify a **default value** if you want:
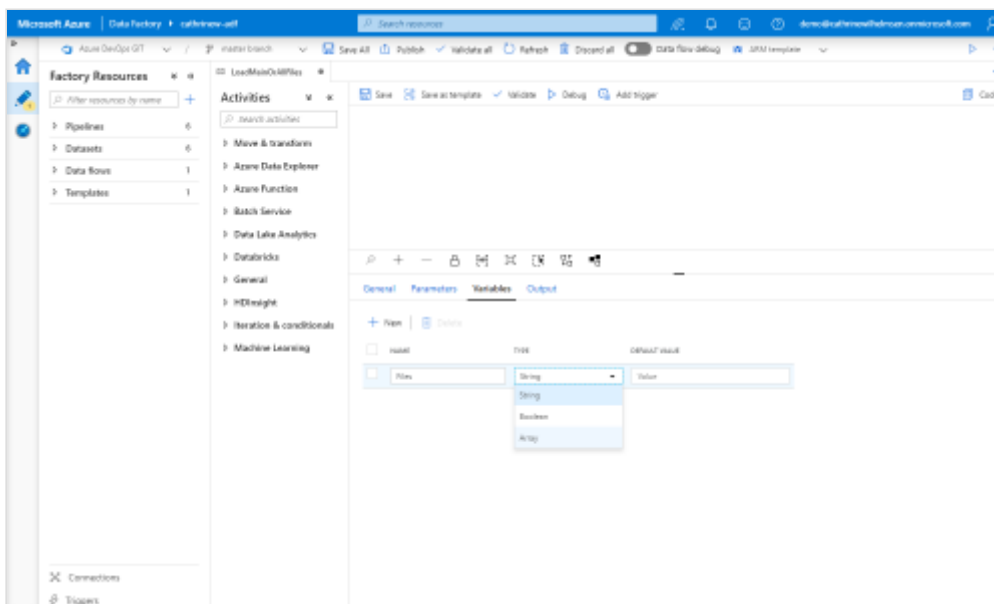
Create two variables. One *array* variable named **Files**, and one *string* variable named **ListOfFiles**:

NAME	TYPE	DEFAULT VALUE
Files	Array ▾	Value
ListOfFiles	String ▾	Value

General   Parameters   **Variables**   Output

+ New | 🗑 Delete

Then, create a *bool* parameter named **LoadAllFiles**:

General   **Parameters**   Variables   Output

+ New | 🗑 Delete

NAME	TYPE	DEFAULT VALUE
LoadAllFiles	Bool ▾	false

🏠   *cathrine*   *adf*   *biml*   *speaking*   Search...   🔍

Set Variable

Add a new **set variable** activity, go to **variables**, and choose the **Files** variable:

Add the array **value** "*colors,inventories,inventory_parts,part_categories,parts,sets,themes*" (*without quotes and spaces*):
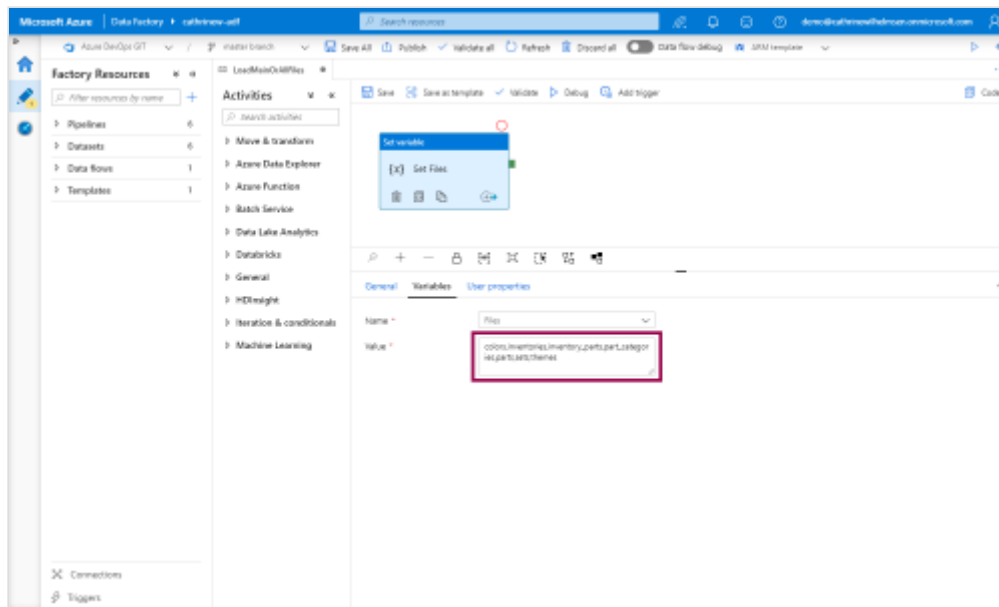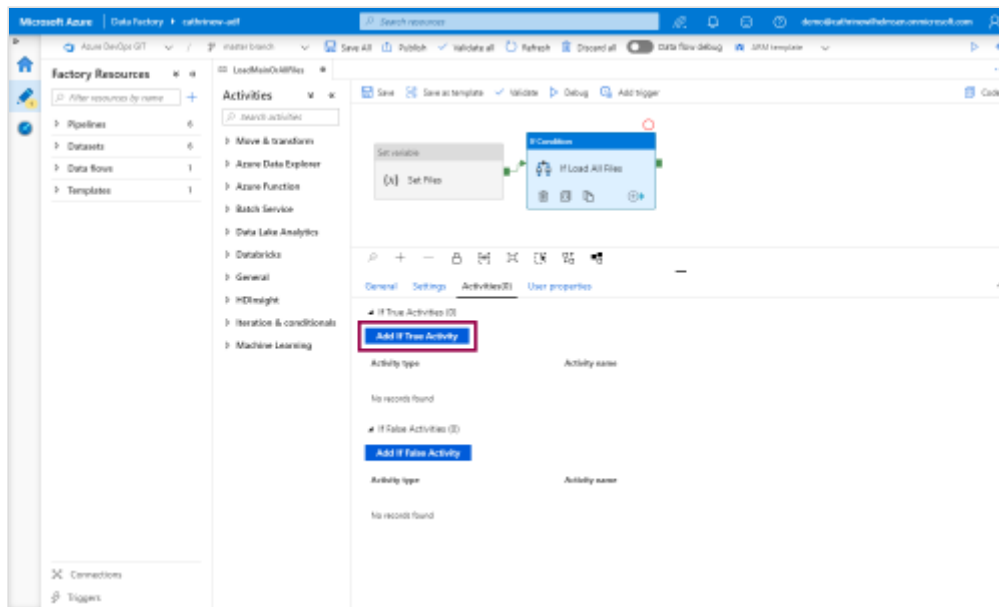
These are the seven Rebrickable files I consider the main files.

Then, add an **if condition**, and use **LoadAllFiles** parameter as the expression. This is a *bool* parameter that will evaluate to *true* or *false*:
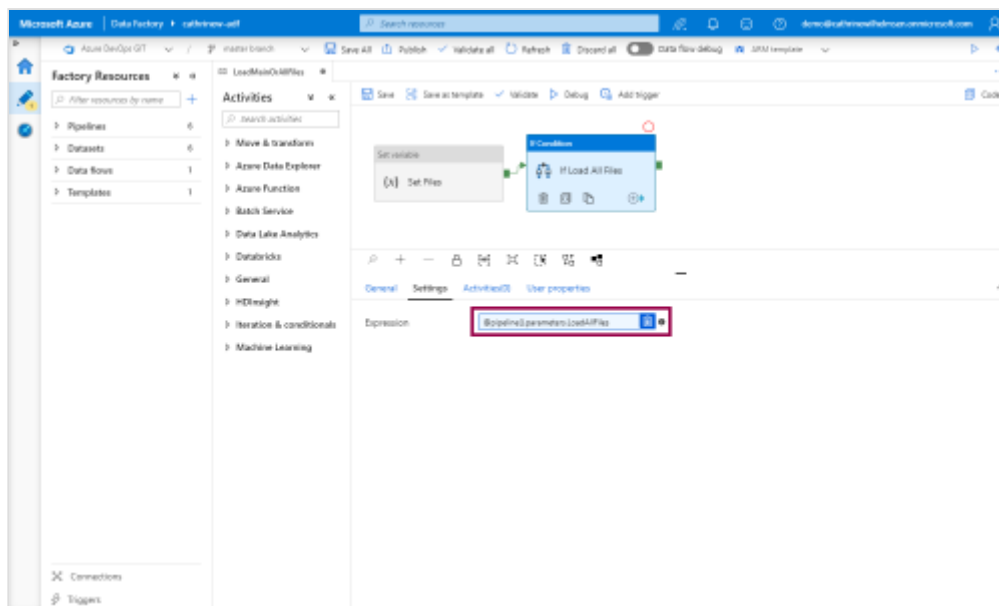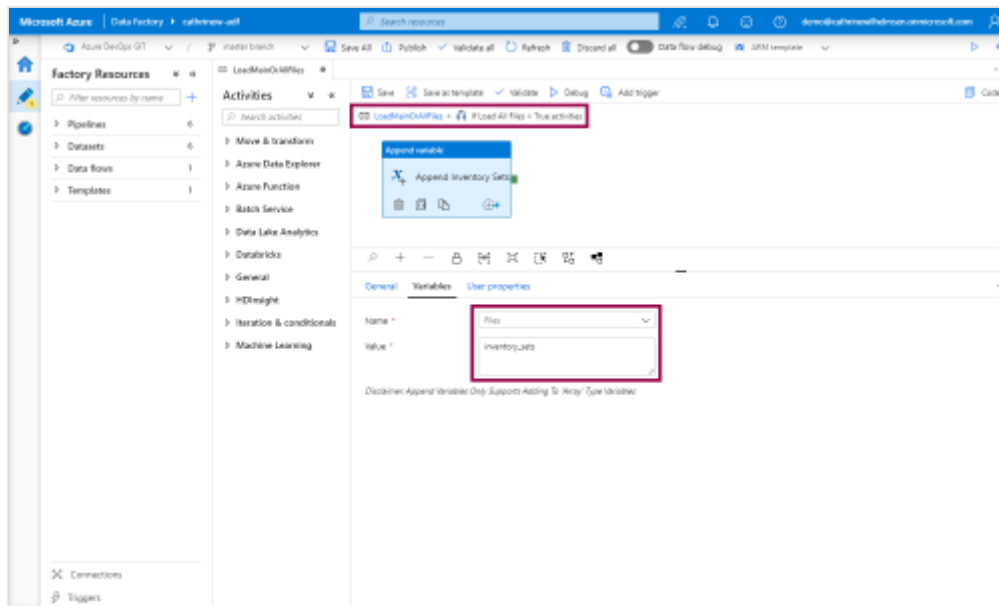

(*Hee hee, I'm sneaking in new activities in this series* ☺)
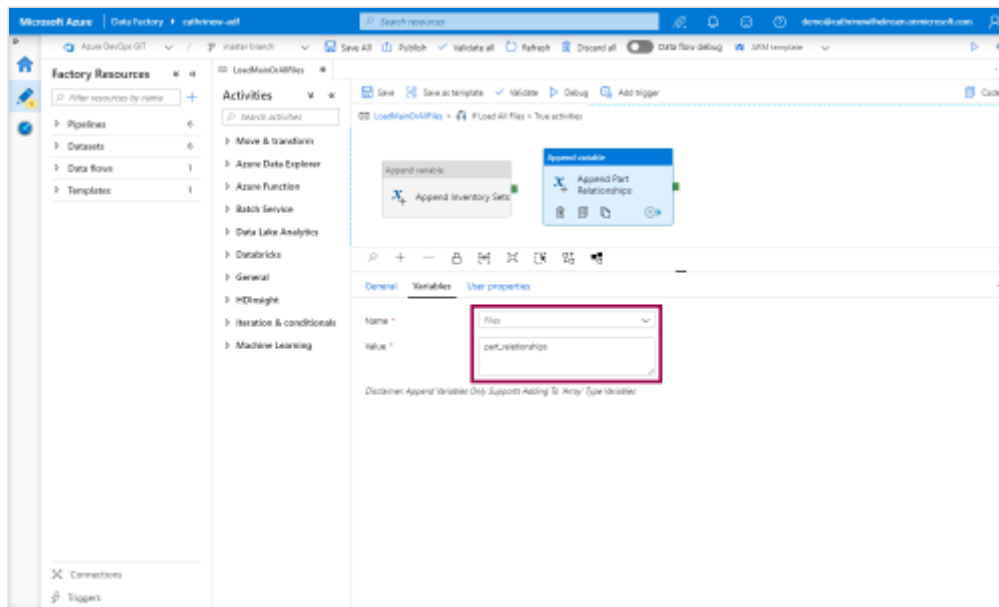
Add an **if true activity**:

Append Variable

Inside the **true activities**, add an **append variable** activity. Choose the **Files** variable, and use the **value** "*inventory_sets*":
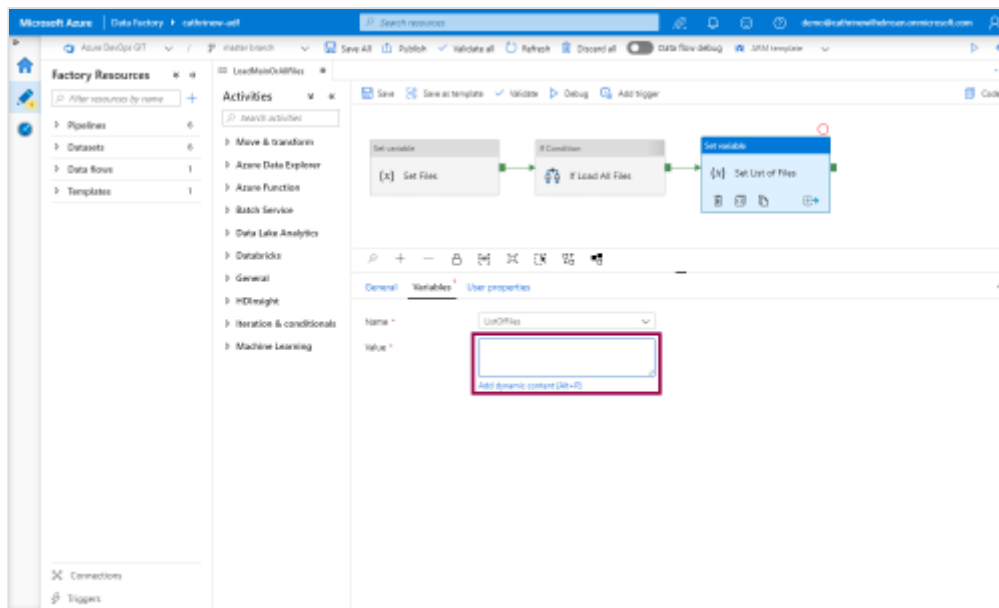

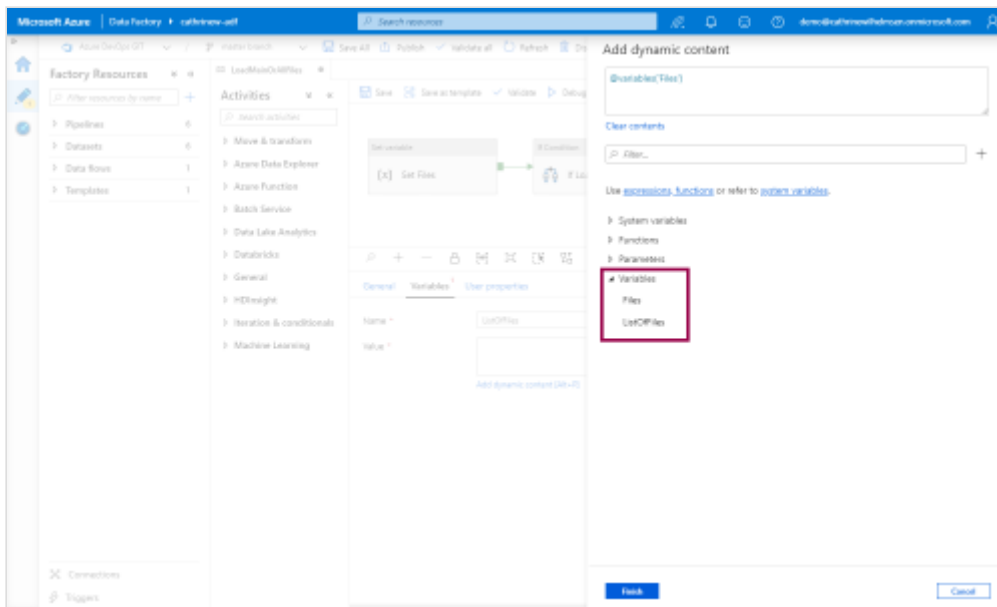Add a second **append variable** activity for "*part_relationships*":

Use Variables

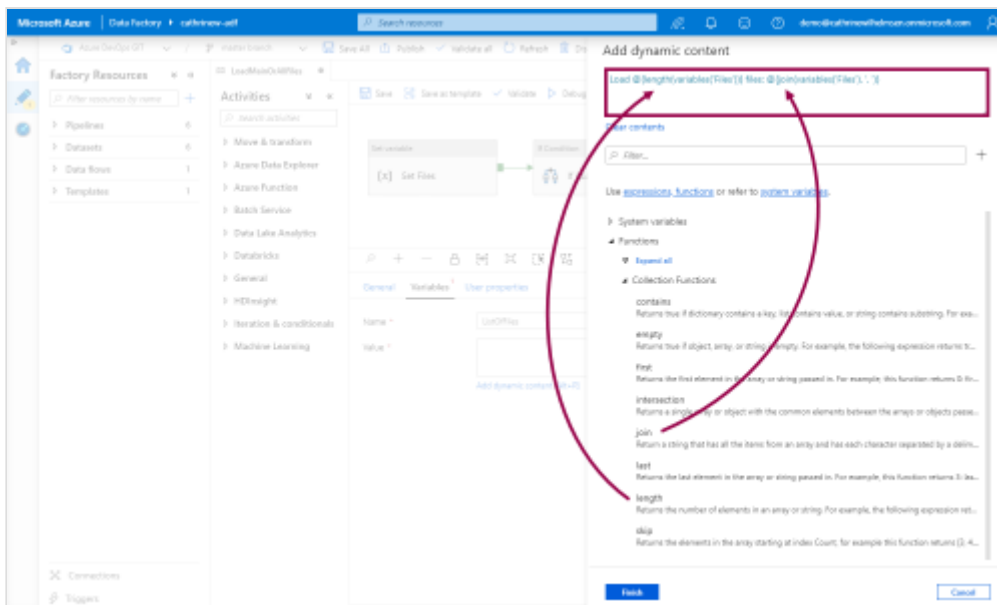Finally, add another **set variable** activity, and click to **add dynamic content**:

In the **add dynamic content** pane, you can click to add a **variable**:

You can use a combination of **variables**, **functions**, and **string interpolation**:

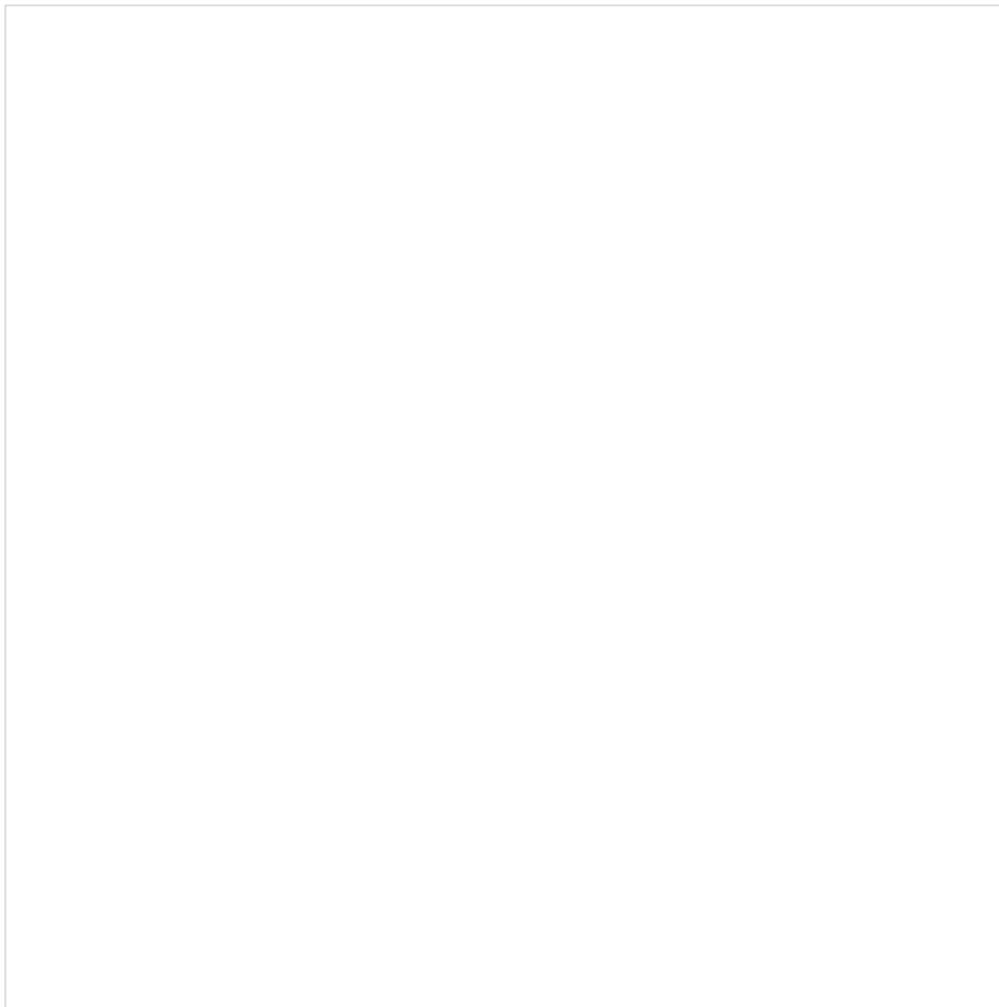Let's test our pipeline! Click **debug**:

And set **LoadAllFiles** to **false**:

Click on the **set files output**. We will see our array with **seven files** listed:

Click on the **set list of Files output**. We will see the **result of the expression** using the variable, functions, and string interpolation. It says "*Load 7 files: colors, inventories, inventory_parts, part_categories, parts, sets, themes*":

Let's try again! **debug** the pipeline again, but this time, set **LoadAllFiles** to **true**:

This, time, we see that the **append variable activities** also ran :) We also get a different **output**:
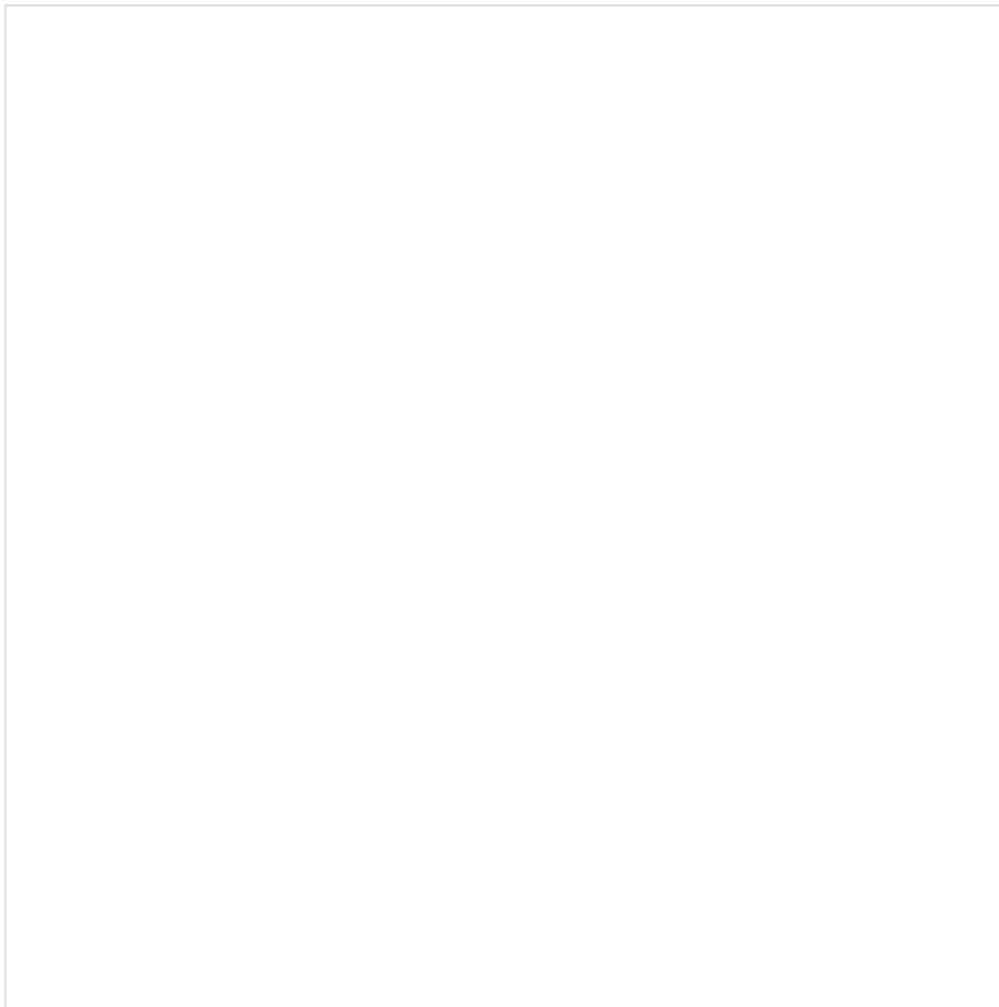
Summary

In this post, we looked at variables and how they are different from parameters. Then we built a pipeline to show how to use the set variable and append variable activities.

We took a sneak peek at working with *arrays* in this post, but we didn't actually *do* anything with it. We just showed the output.

In the next post, however, we will take a closer look at arrays. We will see how they can control a foreach loop!

🤓

← Parameters in Azure Data Factory   ForEach Loops in Azure Data Factory →

Share?

Twitter   LinkedIn   Email   More

Related

[Parameters in Azure Data Factory](#)

[ForEach Loops in Azure Data Factory](#)

[Lookups in Azure Data Factory](#)

🕐 Dec 21, 2019   📁 [Data Platform](#)   🏷 [Azure Data Factory](#)

About the Author

Cathrine Wilhelmsen is a Microsoft Data Platform MVP, BimlHero Certified Expert, Microsoft Certified Solutions Expert, international speaker, author, blogger, and chronic volunteer who loves teaching and sharing knowledge. She works as a Senior Business Intelligence Consultant at Inmeta, focusing on Azure Data and the Microsoft Data Platform. She loves sci-fi, chocolate, coffee, craft beers, ciders, cat gifs and smilies :)

Find me!

🐦 💼 🌐 🎤 📷 ▶️ a ✉️

Subscribe to new posts?

E-mail

Yes, subscribe me!

Recent Posts

[Sneaking back in… (2020 edition)](#)

[Keyboard shortcuts for moving text lines and windows (T-SQL Tuesday #123)](#)

[Speaking at NIC 2020](#)

[Azure Data Factory Training Day at SQLBits 2020](#)

[Personal Highlights from 2019](#)

Popular Posts

[Table Partitioning in SQL Server - The Basics](#)

[Parameters in Azure Data Factory](#)

Preparing for and Taking Microsoft Exam DP-200 (Implementing an Azure Data Solution)

Table Partitioning in SQL Server - Partition Switching

Variables in Azure Data Factory

Top Tags

Azure Data Factory Biml Certifications Don't Repeat Yourself Microsoft Ignite Notepad++ PASS Summit Personal Precon Speaking SQLBits SQLFamily SQLHangout SQLSatOslo SQLSaturday SQL Server SSIS T-SQL Tuesday Volunteering Webinar

All Categories

Select Category ⌄

Full Archive

Select Month ⌄