ThingWorx Flow Connector for Azure Data Lake Storage

User Guide
Version 1.0.0

Software Change Log

Version	Release Date	Changes	Contributors
1.0.0	January 21, 2020	Initial Release	Matt Rice

Introduction and Installation

Extensibility is a core aspect of the architecture and design of ThingWorx. Partners, third parties, and general ThingWorx users can easily add new functionality into the system, seamlessly. Extensions can be in the form of Service (function/method) Libraries, Connector Templates, Widgets, and more. ThingWorx Flow Connectors can be used in a workflow to exchange information between ThingWorx Flow and the connected system, or to trigger workflows based on the occurrence of events in the external system.

This document provides installation and usage instructions for the ThingWorx Flow Connector for Azure Data Lake Storage (ADLS).

About the Connector

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. This connector allows you to interact with and manipulate the unstructured data stored in Azure Data Lake Storage Gen2 including the listing of Containers and Blobs, the creation and deletion of Containers, the upload and download of Blobs, and the deletion of Blobs. The initial release only works with Containers for Blob Storage and do not yet work with File Shares, Tables, or Queues. Authentication is performed via a shared key.

The connector runs in ThingWorx Flow and is built on node.js and utilizes Microsoft's Azure Storage libraries for JavaScript. The connector is intended to work similarly to other storage providers like Box, Dropbox, Google Drive, and OneDrive.

Installing the ThingWorx Flow Connector for Azure Data Lake Storage

There are two ways to install the connector. One is to download a large ZIP file containing all of the necessary node.js modules and the other is to download only the necessary source files and use Node Package Manager (npm) to install the necessary node.js modules.

Install from ZIP File

1. Open a shell or terminal window on the ThingWorx Flow server.
2. Copy the **`dist/ptc-adls-connector-1-0-0.zip`** file from the Git repository to any directory.
3. Unzip the **`ptc-adls-connector-1-0-0.zip`** file.
4. Change into the **`ptc-adls-connector`** directory.
5. Run the following command to deploy the custom connector package to the ThingWorx Flow installation directory:

   ```
   flow-deploy connector --sourceDir . --targetDir
   <ThingWorx_Flow_Install_Directory> --allow_schema_overwrite
   ```

6. Run the following command to load connector metadata into the ThingWorx Flow server:

   ```
   flow-deploy migrate -u <Username_for_ThingWorx_Flow_Database>
   -p <Password_for_ThingWorx_Flow_Database> --
   sourceDir <ThingWorx_Flow_Install_Directory>
   ```

Install from Source

1. Open a shell or terminal window on the ThingWorx Flow server.
2. Download the source code from the **src** directory from the Git repository to any directory.
3. Change into the directory containing the source code.
4. Run the following command to install the required node.js modules:

   ```
   npm install
   ```

5. Run the following command to deploy the custom connector package to the ThingWorx Flow installation directory:

   ```
   flow-deploy connector --sourceDir . --targetDir
   <ThingWorx_Flow_Install_Directory> --allow_schema_overwrite
   ```

6. Run the following command to load connector metadata into the ThingWorx Flow server:

   ```
   flow-deploy migrate -u <Username_for_ThingWorx_Flow_Database>
   -p <Password_for_ThingWorx_Flow_Database> --
   sourceDir <ThingWorx_Flow_Install_Directory>
   ```

For more information about deploying connectors, please see the ThingWorx Help Center at http://support.ptc.com/help/thingworx_hc/thingworx_8_hc/en/index.html#page/ThingWorx/Help/Integration_Orchestration/ConnectorDevelopmentSDK/DeployingConnectors.html

ThingWorx Flow Connector for Azure Data Lake Storage

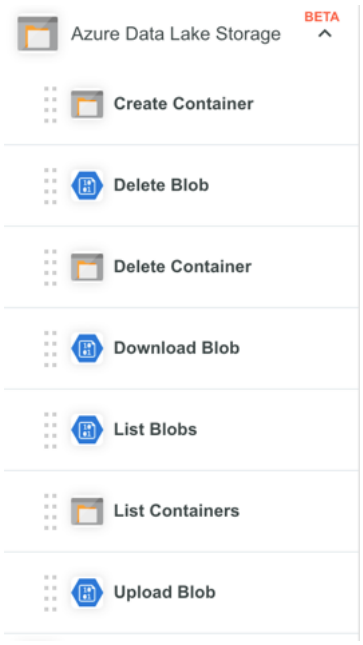
This connector allows you to interact with and manipulate the unstructured data stored in Azure Data Lake Storage Gen2 including the listing of Containers and Blobs, the creation and deletion of Containers, the upload and download of Blobs, and the deletion of Blobs.

Connector Release Version

ADLS Connector 1.0.0.

Supported Actions

- Create Container
- Delete Blob
- Delete Container
- Download Blob
- List Blobs
- List Containers
- Upload Blob



Supported Triggers

None

Supported Authorization

ADLS connector supports the following authorization:

- **Shared Key**

To access the ADLS actions, you need to add a connection.

Adding a Connection

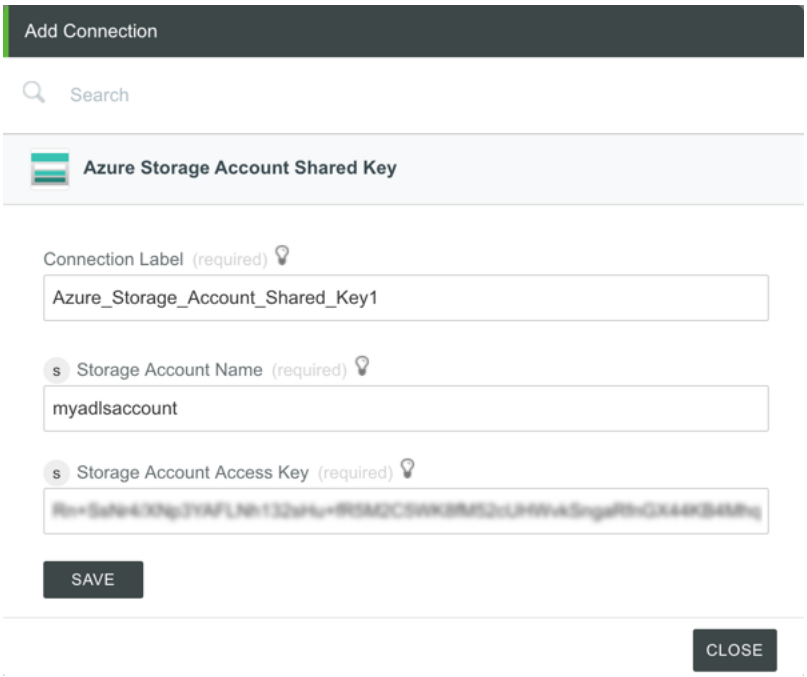
ADLS connection is required for each ADLS connector action. Select one of the following authentication types:

- **Shared Key**

ADLS Shared Key Connection

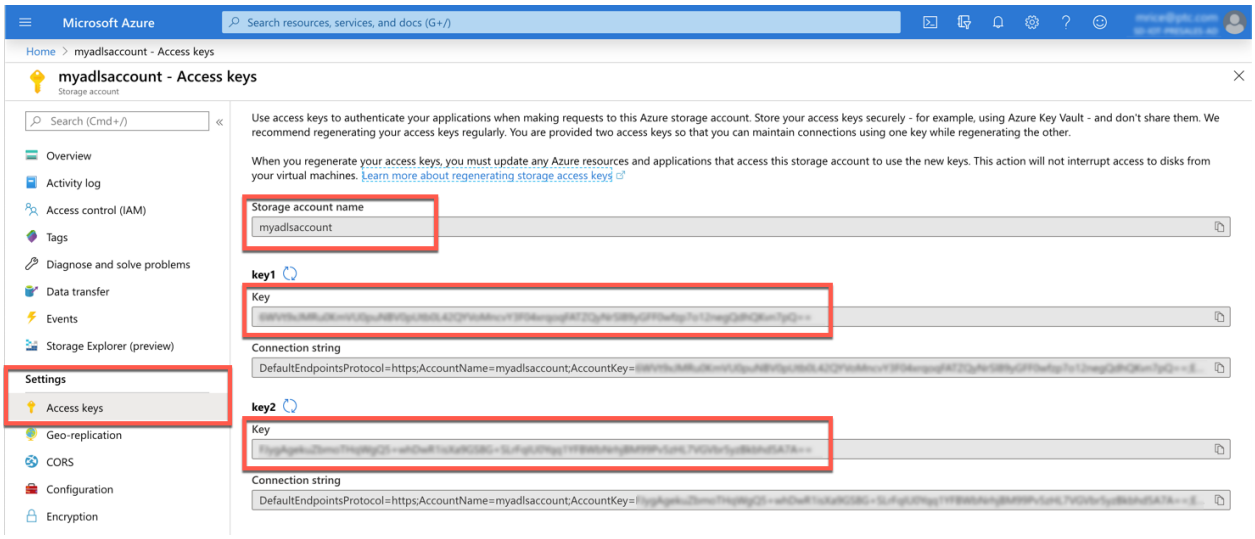
To add a new shared key connection, do the following:

1. From the ThingWorx Flow Dashboard, click on your username in the upper right corner and select **Settings**.
2. Click **CONNECTED ACCOUNTS > Connections**.
3. Click the plus sign and select **Azure Storage Account Shared Key** then enter the following details in the Add Connection window:

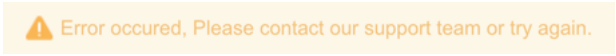


- **Connection Label** – Auto populated. You can provide a connection label of your choice.
- **Storage Account Name** – The name of the Azure storage account. This is the **Storage account name** listed in Settings > Access Keys in the Azure portal for your storage account.
- **Storage Account Access key** – The shared key of the Azure storage account. This is the **Key** value for either key1 or key2 listed in Settings > Access Keys in the Azure portal for your storage account.

> You need to log in to your Azure portal to retrieve this information.



4. Click **SAVE** to save and test the authorization.
    a. Testing the authorization should happen within 1-2 seconds and will timeout after 1 minute. If it fails you will get an error and you cannot save the connection.



    b. A successful authentication test will save the connection.

5. Click **CLOSE** to close the Add Connection window.

Click **TEST** to verify the authentication.

Create Container

Use this action to create a Container.

Using the Create Container Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **Create Container** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The Create Container action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container to create.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

5. Click **Done**.

Delete Blob

Use this action to delete a Blob.

Using the Delete Blob Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **Delete Blob** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The Delete Blob action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container that contains the blob.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

5. In the **Blob Name** field, enter the name of the blob to delete.
6. Click **Done**.

Delete Container

Use this action to delete a Container.

Using the Delete Container Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **Delete Container** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The Delete Container action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container to delete.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

5. Click **Done**.

Download Blob

Use this action to download a Blob.

Using the Download Blob Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **Download Blob** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The Download Blob action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container that contains the blob.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

5. In the **Blob Name** field, enter the name of the blob to download.
6. Click **Done**.

List Blobs

Use this action to list blobs in a container.

Using the List Blobs Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **List Blobs** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The List Blobs action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container from which to get the list of blobs.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

5. Click **Done**.

List Containers

Use this action to list containers.

Using the List Containers Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **List Containers** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The List Containers action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. Click **Done**.

Upload Blob

Use this action to upload a blob.

Using the Upload Blob Action

To use this action in your workflow, you need to connect it to ThingWorx Flow. To connect to the flow, do the following:

1. Drag the **Upload Blob** action under the **Azure Data Lake Storage** connector to the canvas, place the point on the action, and then click ⚙ or double-click the action. The Upload Blob action window opens.
2. Edit the label name, if needed. By default, the label name is the same as the action name.
3. Specify an existing **Shared Key Connection** or add a new one.
4. In the **Container Name** field, enter the name of the container to store the blob.

> The Container and Blob names must comply with the naming conventions specified at https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata.

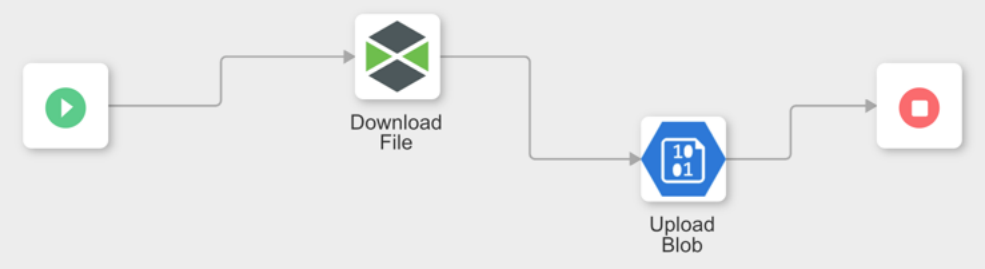
5. In the **Blob Name** field, enter the name of the blob.
6. In the **Data** field, enter the data for the blob. This can be plain text or encoded binary data.
7. In the **Encoding** field, select the encoding type of the data.
8. Click **Done**.

Examples

Here are examples of using the connector for various use cases.

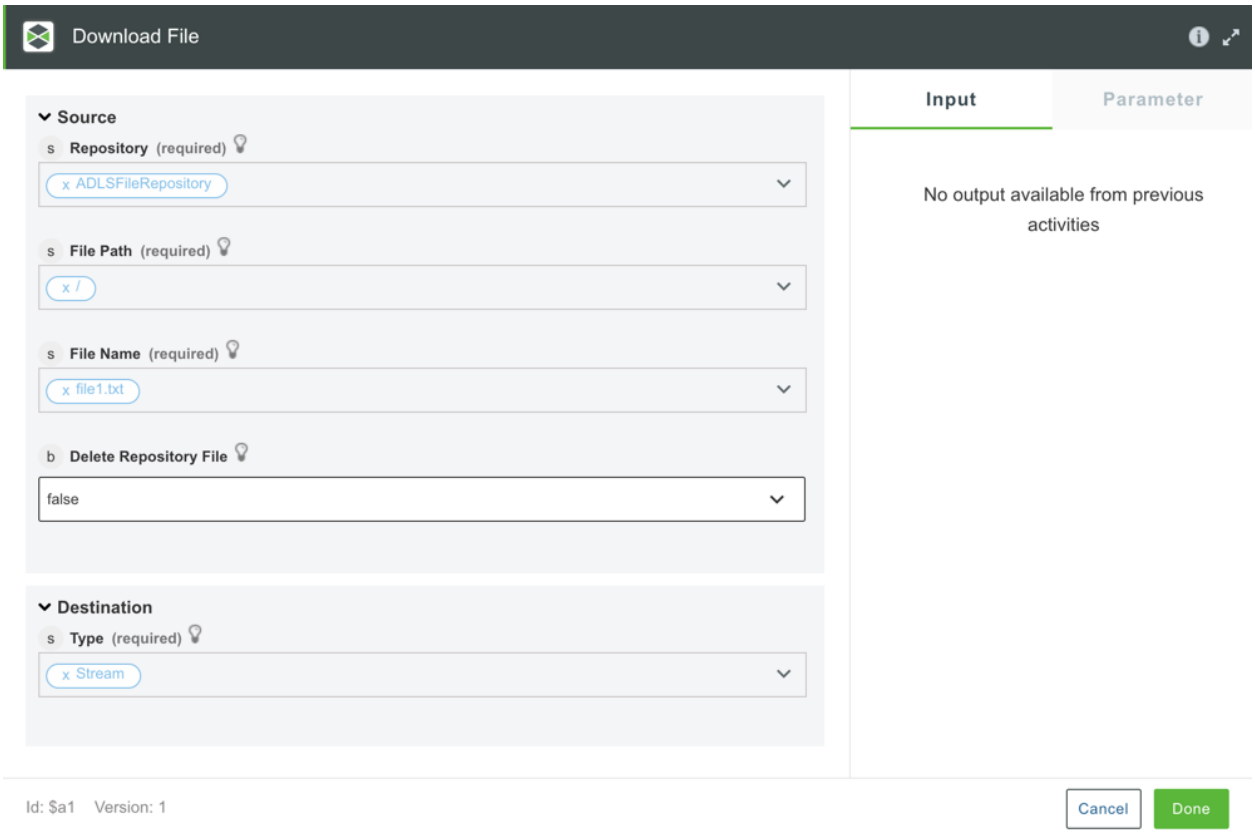
Upload a Blob from ThingWorx File Repository

Create a workflow using the ThingWorx Download File action and the ADLS Upload Blob action.



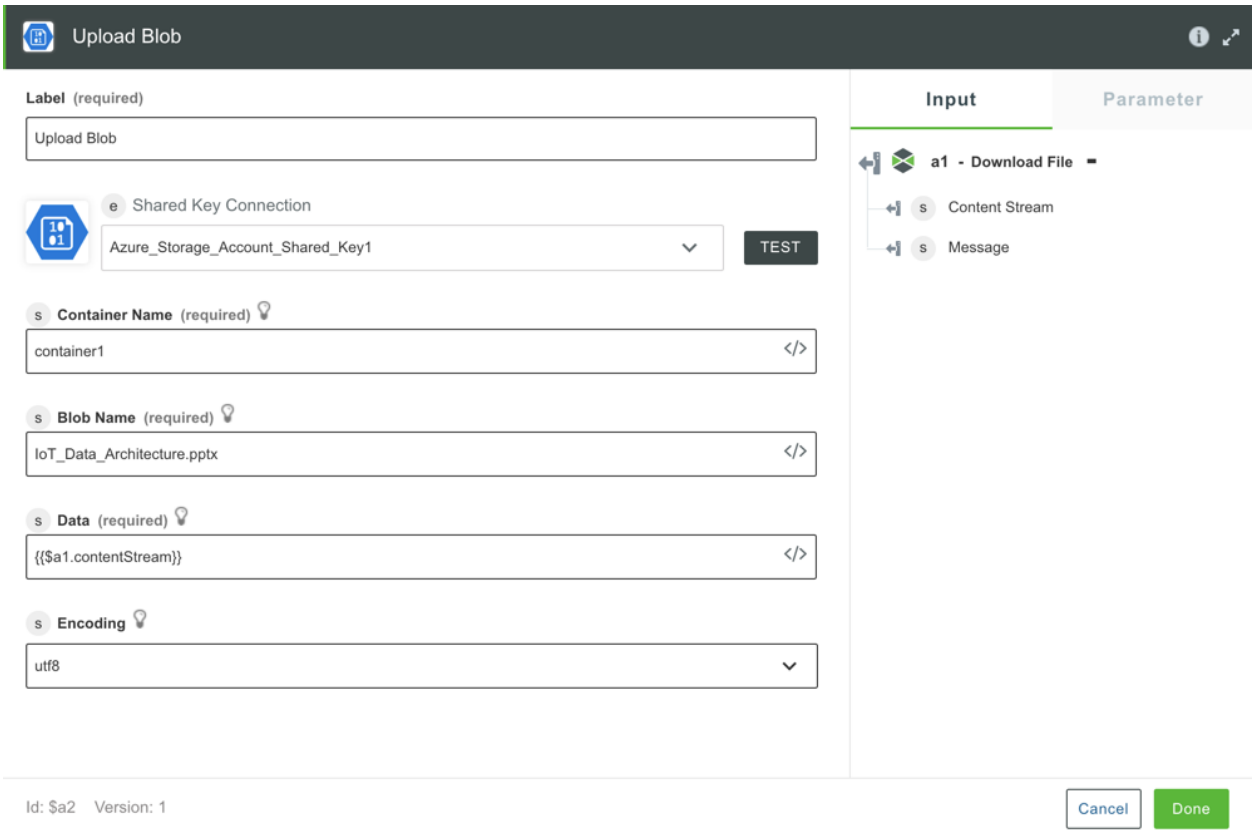
Configure the **Download File** action:

1. Select the **Authentication Type** and **Connection**.
2. In the **Source** field, select the following source file details:
    - **Repository** – ThingWorx source repository from where the file is to be downloaded.
    - **File Path** – Path within the repository.
    - **File Name** – Name of the repository file to be downloaded.
    - **Delete Repository File** – Select one of the following actions:
        - **false** – Does not delete the file from the ThingWorx repository after download.
        - **true** – Deletes the file from the ThingWorx repository after download.
3. In the **Destination** field, select **Stream** to stream the binary content of the file to the Upload Blob action.



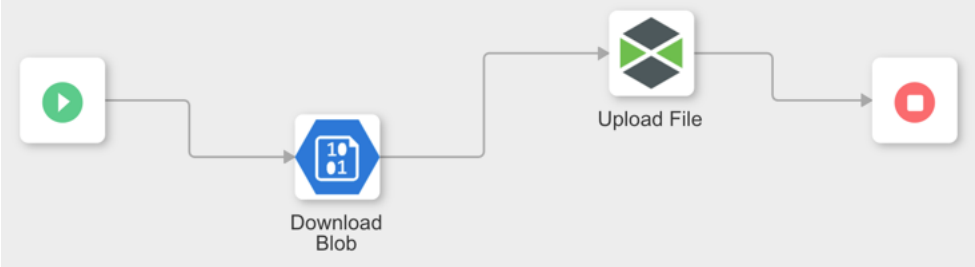
Configure the **Upload Blob** action:

1. Specify an existing **Shared Key Connection** or add a new one.
2. In the **Container Name** field, enter the name of the container to store the blob.
3. In the **Blob Name** field, enter the name of the blob.
4. In the **Data** field, drag the Content Stream property from the Download File output into this field. It will be something like `{{$a1.contentStream}}`.
5. In the **Encoding** field, select the encoding type of the data.



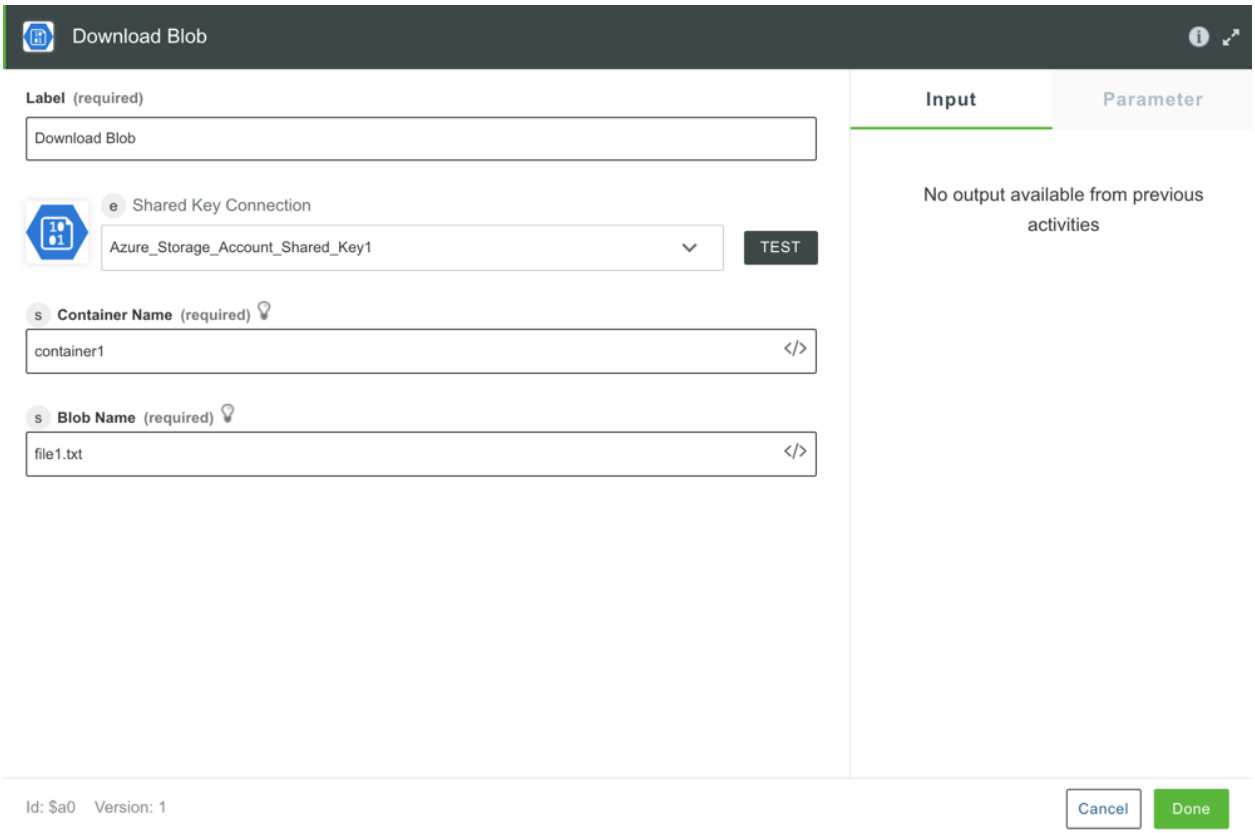
Download a Blob to ThingWorx File Repository

Create a workflow using the ThingWorx Download File action and the ADLS Upload Blob action.



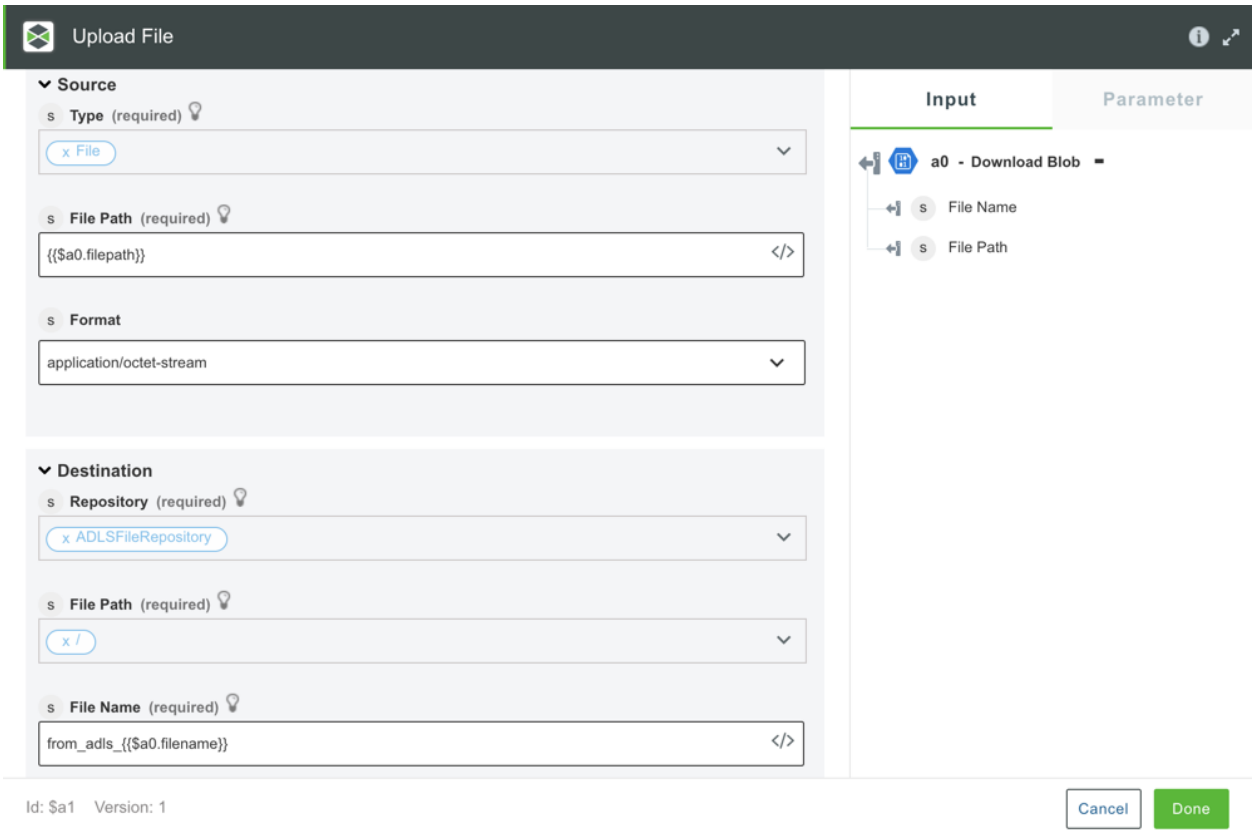
Configure the **Download Blob** action:

1. Specify an existing **Shared Key Connection** or add a new one.
2. In the **Container Name** field, enter the name of the container that contains the blob.
3. In the **Blob Name** field, enter the name of the blob to download.



Configure the **Upload File** action:

1. Select the **Authentication Type** and **Connection**.
2. In the **Type** field, select a format in which the source data is provided.
   - **File**
     - **File Path** – The File Path property from the Download Blob output. For example, `{{$a0.filepath}}`.
     - **Format** – File content format.
3. In the **Destination** field, enter the destination details where the data is uploaded:
   - **Repository** – Select the ThingWorx repository where the file is to be uploaded.
   - **File Path** – File path within repository.
   - **File Name** – Name of the repository file. For example, you can use the File Name property from the Download Blob output such as `{{$a0.filename}}`.


Upload File
Source
Type (required)
File
File Path (required)
{{$a0.filepath}}
Format
application/octet-stream
Destination
Repository (required)
ADLSFileRepository
File Path (required)
/
File Name (required)
from_adls_{{$a0.filename}}
Input
Parameter
a0 - Download Blob
File Name
File Path
Id: $a1
Version: 1
Cancel
Done

Known Limitations

Only Storage Containers and Blobs are currently supported. File Shares, Tables, and Queues are not yet supported.
Hierarchical Containers have not been tested.

Compatibility

This extension was tested for compatibility with the following ThingWorx Flow version(s):

ThingWorx Flow Version	8.5.0, 8.5.1, 8.5.2, 8.5.3

Document Revision History

Revision Date	Version	Description of Change
January 17, 2020	1.0.0	Initial version