

Publish to Azure Blobs How to Configure

Complete the following steps in order to configure the Azure Blobs Publisher.

On this page

- [Step 1. Launch Aspire](#)
- [Step 2. Open the Content Source Management Page](#)
- [Step 3. Add a New Content Source](#)
- [Step 4. Add a New Publish to Azure Blobs to the Workflow](#)
 - [Step 4a. Specify Publisher Information](#)

Step 1. Launch Aspire

Launch Aspire (if it's not already running). See: [Launch Control](#)



Step 2. Open the Content Source Management Page

Browse to: <http://localhost:50505>.

For details on using the Aspire Content Source Management page, please refer to [Admin UI](#).

Step 3. Add a New Content Source

For this step, please follow the procedure from the Configuration Tutorial of the connector of your choice. Refer to the [Connector list](#)

Step 4. Add a New Publish to Azure Blobs to the Workflow

To add a Publish to Azure Blobs drag from the **Publish to Azure Blobs** rule from the *Workflow Library* and drop to the *Workflow Tree* where you want to add it.

This will automatically open the **Publish to Azure Blobs** window for the configuration of the publisher.

Step 4a. Specify Publisher Information

In the Publish to Azure Blobs window, specify the connection information needed to publish to Azure Blob storage.

1. **Storage Connection String:** This is the connection string for the Azure Blob storage service, which contains four parts.

- Default Endpoints Protocol, with two possible values: http or https. For example: "DefaultEndpointsProtocol=https;"

- Account Name, which is the name of your Microsoft Azure storage account. For example: "AccountName=myAccount;"

Publish to Azure Blob



Coordinates: com.accenture.aspire:app-azure-blob-publisher

Name: Publish to Azure Blob

Storage Connection String: DefaultEndpointsProtocol=http;AccountName=myAccount;AccountKey=myKey;BlobEndpoint=http: ?

Blob Container Name: defaultcontainer ?

☒ Clean container before full crawl ?

Upload binary objects: ☒ ?

Binary Objects

☒ Use only one extension ?

Unique binary file extension:

obj ?

File extension exclude list:

jpg,jpeg,gif,png,tif,mp3,mp4,mpg,mpeg,avi,mkv,wav,bmp,swf,war,rar,tgz,dll,exe,class,docx,pdf ?

Use Transformation file: ☒

JSON Transformation: S{appbundle.home}/config/groovy/aspireToBlob.groovy ?

Soft Delete: ☒

Delete Flag Name: Deleted ?

Add Blob Metadata: ☒

Blob Metadata

Metadata Source: Original Document ?

Add New

Metadata Name:

Field required. ?

Metadata Value:

Field required. ?

Default Value:

? ?

Add

Cancel

- **BlobEndpoint**, indicates the URL for the blob storage repository. For example: "BlobEndpoint=<http://mystorageaccount.blob.core.windows.net>"
2. **Blob Container Name**: Enter the name of the container inside the Azure Blob storage where you want to publish your results.
 3. **Clean container before full crawl**: Mark this option if you want to clean the blob container before a full crawl.
 4. **Upload binary objects**: Mark this option if you want to upload the binary objects together with the json objects. For the Binary Objects to be successfully uploaded you need to disable the extract text option in the Connector settings.
 5. **Use only one extension**: Mark this option if you want all the binary objects to be uploaded using the same file extension. The file extension to be used is specified in the next field "*Unique binary file extension*". If the option is not marked, then the uploaded files will keep their original file extension.
 6. **Unique binary file extension**: if the option "*Use only one extension*" is marked, then this is the used file extension for all uploaded binary objects.
 7. **File extension exclude list**: This is a comma separated list for all the file extensions you want to exclude from the upload binary process. Add the extensions of the files that you don't want to be uploaded into this list.
 8. **Use Transformation file**: Select the check box to use a Groovy script (indicated in the JSON Transformation field) to transform the content and store the results of the transformation.
Clear the check box to create the standard Aspire JSON content.
 9. **JSON Transformation**: Enter the location of the file containing the JSON Transformation script. This script will be used to transform the data from Aspire as it is posted to the Azure Blob Container.
See [JSON Transformation Script](#) for details on the format of the script.
 10. **Soft Delete**: Select the check box to execute a soft delete (object mark as deleted) rather than a physical delete from Azure Blob.
 11. **Delete Flag Name**: Enter the name of the flag used to mark document as deleted in Azure Blob. The flag will be stored as part of the blob metadata (Deleted=true).
 12. **Add Blob Metadata**: Select the check box to add metadata to the Azure Blob.
 13. **Metadata Source**: Specify where to take metadata values from. Select *Original Document* from the drop down if the values for the metadata fields are stored inside the original input document (Aspire Doc) or select *Transformed Document* if the values for the metadata fields are stored inside the transformed output document.
 14. **Metadata Name**: Enter the name of the blob metadata field. If a name specified as a metadata field matches the delete flag name used for soft deletes, the value will be overwritten by the soft delete handler. This is a required value.
 15. **Metadata Value**: Enter the field name or path (e.g. connectorSpecific/Author) inside Aspire or transformed document where to take the value from. This is a required value.
 16. **Default Value**: Enter the default value to be used when the metadata value does not exist.
 17. **Debug**: Select the check box to run the publisher in Debug mode.
 18. Click **Add**.

Once you've clicked **Add**, it will take a moment for Aspire to download all of the necessary components (the Jar files) from the Maven repository and load them into Aspire. Once that's done, the publisher will appear in the Workflow Tree.



For details on using the Workflow section, please refer to [Workflow](#) introduction.