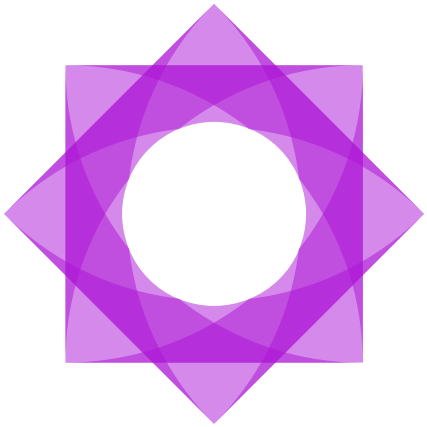


Lasernet 10.

Lasernet Azure 10

Torben Pedersen, Sunil Panchal, Adam McStravick
Revision 9
2026-01-27

Formpipe.

Formpipe.

Contents.

1 Introduction.

1.1 Who Should Use This Guide?

This guide is written for Lasernet Developers. It is intended primarily as a reference to the different Microsoft Azure Storage functions in Lasernet.

It provides the information required for successfully integrating Microsoft Azure Storage and Lasernet in your business.

2 Terms of Use.

No part of this publication may be reproduced, transmitted, transcribed, or translated into any language in any form by any means without the prior written permission of Formpipe Software. The information in this manual is subject to change without notice. Any company names or data is fictive unless otherwise stated.

Formpipe Software shall not be liable for any loss or damage whatsoever arising from the use of this manual and the information contained therein (including errors or omissions).

Trademarks of other companies mentioned in this document appear for identification purposes only and are the property of their respective companies.

3 Azure.

3.1 Microsoft Azure

Microsoft Azure is a cloud computing platform and infrastructure created by Microsoft for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

Lasernet gives access to a limited subset of services in Azure – namely the Storage and Service Bus. Storage is only partly supported, as Lasernet only interfaces with BLOB and Queue.

Azure **Storage Blob** is a service that stores unstructured data in the cloud as objects/blobs. Blob storage can store any type of text or binary data, such as a document, media file, or application installer. Blob storage is also referred to as object storage.

Azure **Storage Queue** is a service for storing large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. A single queue message can be up to 64 KB in size, and a queue can contain millions of messages, up to the total capacity limit of a storage account.

Common uses of Storage Queue include:

- Creating a backlog of work to process asynchronously.
- Passing messages from an Azure web role to an Azure worker role.

The maximum time that a message can remain in the queue is 7 days.

Azure **Service Bus** is a service that provides a multi-tenant service for connecting application through the cloud. A single service by message can be up to 256 KB or 1 MB in size depending on your subscription.

This manual assumes user familiarity with Microsoft Azure services and focuses only on configuring Lasernet to interface with the Storage Queue and Service Bus. For more information about Microsoft Azure Services please visit the Azure Website (https://azure.microsoft.com).

3.2 Lasernet Components Overview

Lasernet for Microsoft Azure consists of the following components:

3.2.1 Azure AD Auth Modifier (Microsoft Entra ID)

The Azure AD Auth modifier logs into Microsoft Entra ID for a given user and returns an access token, which can be used to access different services in Azure.

3.2.2 Azure Hybrid Connection Input Module

The Azure Hybrid Connection input module provide an easy and convenient way to connect the Web Apps feature in Azure App Service and the Lasernet Azure SAS Auth output module to a Lasernet Server running on-premises behind the firewall.

3.2.3 Azure SAS Auth Modifier

Azure SAS Auth Modifier Shared Access Signatures (SAS) is the primary security mechanism for Service Bus messaging. It exchanges a connection string with a token to be used with either the SOAP Web Service module or the HTTP module. With the Azure SAS Auth modifier, Lasernet can both act as client (Azure SAS Auth) and server (Azure Hybrid).

3.2.4 Azure Storage Input Module

The Azure Storage input module enables Lasernet Server to retrieve files from BLOB storage or receive messages sent via a Queue.

3.2.5 Azure Storage Output Module

The Azure Storage output module enables Lasernet Server to store files in BLOB storage or send messages via a Queue.

3.2.6 Azure Storage Connection

An Azure Storage connection is used by an Azure Storage Command to authenticate Lasernet with Microsoft Azure Storage.

3.2.7 Azure Storage BLOB Command

The Azure Storage BLOB command is able to perform one of five commands:

- Create Container
- Delete container
- Upload BLOB
- Download BLOB
- Delete BLOB

3.2.8 Azure Storage Queue Command

The Azure Storage Queue command is able to perform one of five commands:

- Create Queue
- Delete Queue
- Insert Message
- Peek Message
- Get Message

3.2.9 Scripting

Script support enables you to batch call your Azure Storage commands in sequence.

3.2.10 Azure Service Bus Input Module

The Azure Service Bus input module enables Lasernet Server to receive messages from a Service Bus Queue or a Topic/Subscription.

3.2.11 Azure Service Bus Output Module

The Azure Service Bus output module enables Lasernet Server to post messages to a Service Bus Queue or Topic.

3.3 Troubleshooting and Support

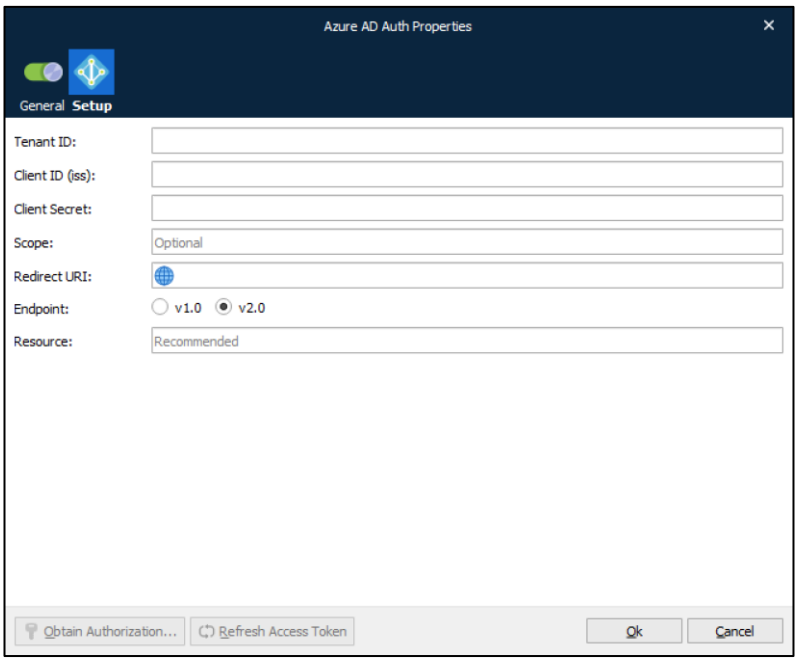
For technical support please contact your local partner or reseller.

Email help can be reached via support.lasernet@formpipe.com

4 Reference.

4.1 Azure AD Auth Modifier (Microsoft Entra ID)

The **Azure AD Auth** modifier logs into Microsoft Entra ID for a given user and retrieves a token which can be used to access different services in Azure. This could be SharePoint, Exchange, or custom web services developed by third parties, which you want Lasernet to communicate with. It can also request different scopes, which determine what Lasernet can access, if granted.



Tenant ID	The ID of the AAD directory in which you created the application.
Client ID (iss)	Credentials (ID) for Microsoft Entra ID Authentication.
Client Secret	Credentials (Secret key) for Microsoft Entra ID Authentication.
Scope	The request may have one or more scope values indicating additional access requested by the application. The authorization server will need to display the requested scopes to the user. The setting is optional.
Redirect URI	The redirect URI is the URL within your application that will receive Microsoft Entra ID credentials.
Endpoint	Select between Endpoint version v1.0 or v2.0.
Resource	Shows the ID of the API that the modifier wants to access on behalf of the user.

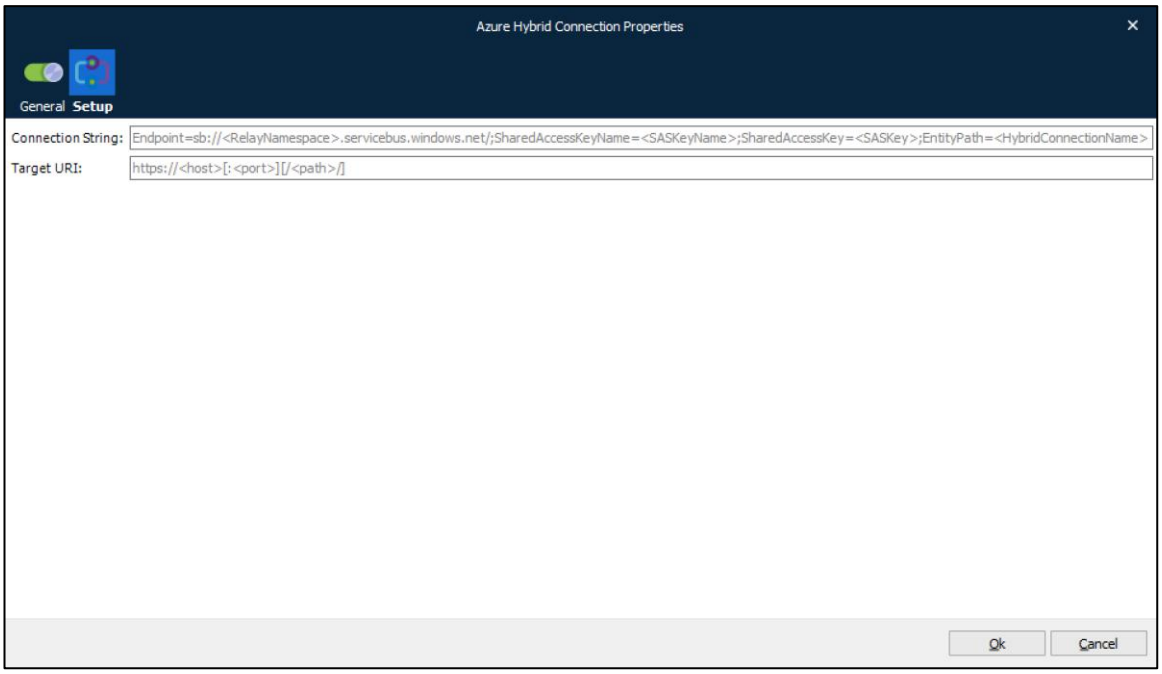
4.1.1 JobInfos

The Azure AD Auth modifier (Microsoft Entra ID) sets a JobInfo.

OAuth2AccessToken Value of the OAuth 2.0 access token.

4.2 Azure Hybrid Input Module

The Azure Hybrid Connection Input module provides an easy and convenient way to connect the Web Apps feature in the Azure App Service, or the Lasernet HTTP Output module (with Azure SAS Authentication), to a Lasernet Server running behind a firewall.

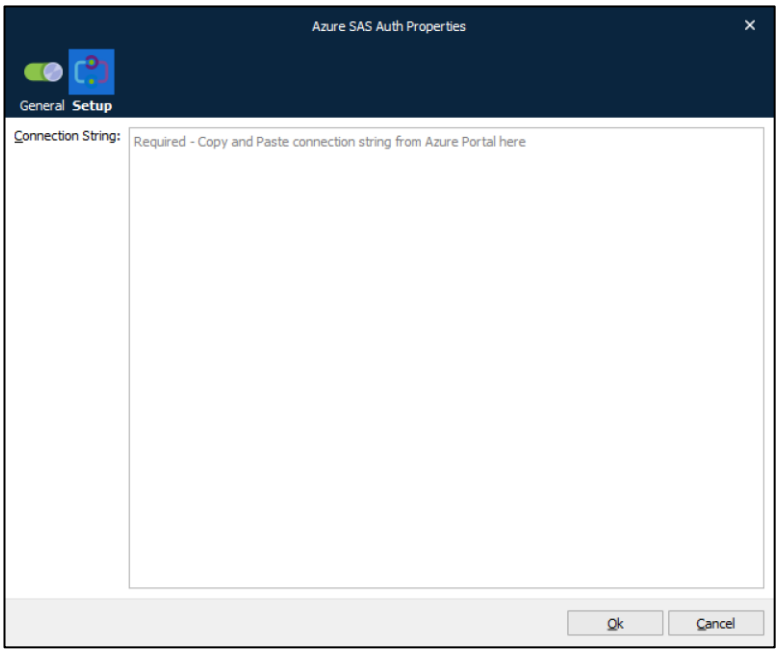


Connection String Locate and copy/paste your connection string from the Azure Portal.

Target URI The Target URI is the string that identifies your logical or physical resource, beginning with the HTTP protocol followed by the host name, port number and the path.

4.3 Azure SAS Auth Modifier

Shared Access Signatures (SAS) are the primary security mechanism for Service Bus messaging. They exchange a connection string with a token that is used in the SOAP Web Service module or the HTTP module. With the Azure SAS Auth modifier, Lasernet can both act as client (Azure SAS Auth) and server (Azure Hybrid).



Connection String Locate and copy/paste your connection string from the Azure Portal.

4.3.1 JobInfos

The Azure SAS Auth modifier sets a JobInfo.

ServiceBusAuthorization Value of the Azure SAS authentication token.

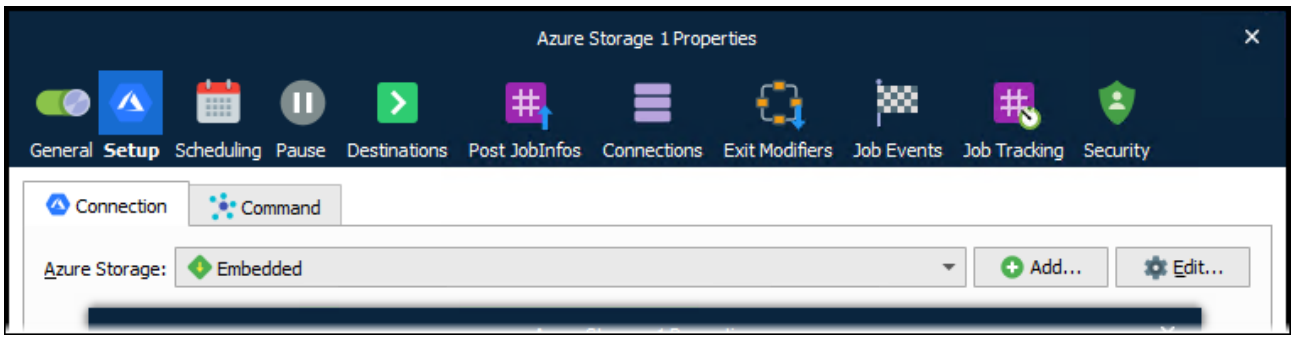
4.4 Azure Storage Input Module

The Azure Storage input module is used to pull files from BLOB Storage or messages from a Queue, at a defined interval via the Scheduler.

The setup of the module consists of two pages: **Connection** and **Command**.

4.4.1 Connection

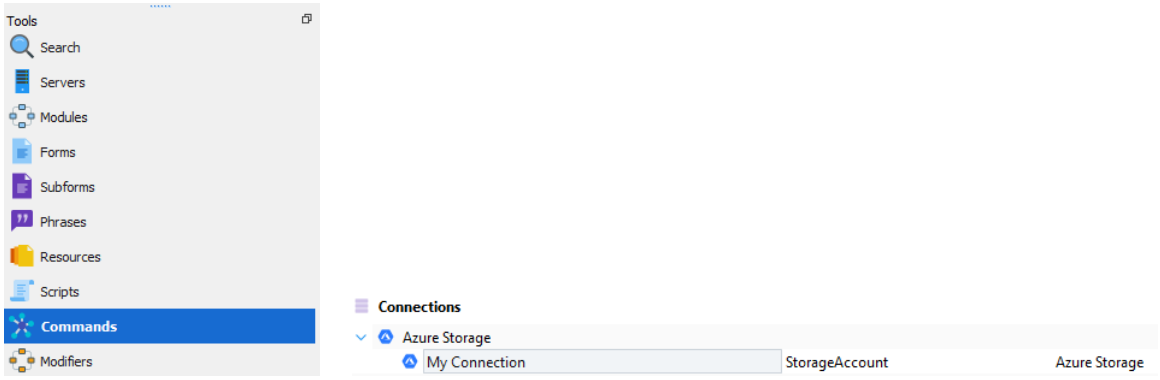
In order to access the Azure Storage API, the module's **Connection** properties must be correctly configured.



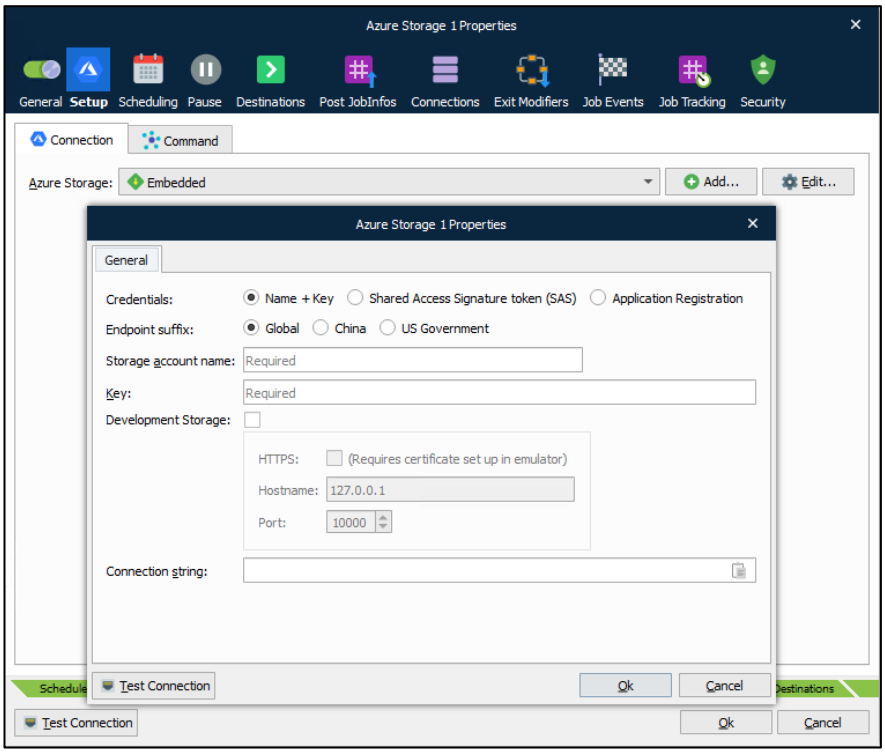
On the **Connection** tab, you have the following options:

- **Create an "embedded" connection:** Embedded connection settings apply only to this module.
  - To do this, select **Embedded** from the **Azure Storage** list, then click **Edit** to configure the connection.
- **Create a new global connection:** Multiple modules in this Lasernet configuration can share the same "global" Connection item (to centralize management of connection configuration).
  - To create a global **Connection** item, click **Add** beside the **Azure Storage** list.
- **Use an existing global connection**
  - To use an existing global **Connection** item, select it from the **Azure Storage** list.
  - To edit its connection properties, click **Edit**.

The image below shows a global **Connection** item in the **Commands** area of Lasernet Developer.



For both embedded and global connections, Lasernet displays a connection **Properties** window when you edit the connection.



The module can use one of the following authentication methods:

- **Name + Key:** Access key authentication. Requires a **Storage account name** and a **Key**.
- **Shared Access Signature (SAS):** A signed URI that grants access to a specific Azure Storage resource.
- **Application Registration:** Authentication based on managed application identities in Entra ID. Requires a **Client ID** and **Client Secret** (which are the application credentials that Lasernet will use to authenticate with Entra ID).

Each of these authentication methods has specific connection properties (described in the tables below).

Note: Global connections also have **Name** and **Description** properties to describe the Connection item itself.

Note: You can use JobInfo substitution in any connection property value (including **SAS Token**).

Tip: After you configure the connection, click **Test Connection** (in the bottom corner) to test the connection properties.

4.4.1.1 Name + Key

Setting	Description
Endpoint suffix	Select the Microsoft cloud that hosts the storage account. Microsoft maintains separate clouds for data residency and regulatory compliance reasons.
Storage account name	All access to Azure Storage is done through a storage account. This account can be a general-purpose storage account or a Blob storage account specifically for storing objects/blobs.
Key	The access key used to authenticate Lasernet while calling commands.
Development Storage	If selected, the module connects to an Azure Storage emulator, which (for testing purposes) avoids the need for a real Azure Storage account. To use this feature, install an emulator such as Azurite, which Lasernet can use locally or remotely: • You can install Azurite on any server or workstation that will run or load the Lasernet configuration that you want to test. For example, you might install it on Lasernet test servers or computers running Lasernet Developer. By default, for security reasons, Azurite accepts only local requests (to 127.0.0.1). • Or, Azurite can be configured to accept remote connections, so that it is shared. If you do this, ensure that Lasernet can communicate with Azurite over the network. After installing Azurite, configure and run it. For more information about installing, configuring, and running Azurite, see Microsoft documentation. When **Development Storage** is selected, Lasernet generates a connection string that causes Lasernet to send Azure Storage requests to the emulator. If necessary, modify the following settings: • **HTTPS:** Specifies that an HTTPS connection to the emulator must be established. ○ If **HTTPS** is selected, you must configure the emulator to use a certificate. • **Hostname:** The hostname or IP address of the emulator. The default value (127.0.0.1) is equivalent to localhost. • **Port:** The port that the emulator is listening on.
Connection string	If you copy a connection string for a storage account to the clipboard, this **Connection string** property can interpret it and use its content to complete the **Storage account name** and **Key** properties. Click the **Parse connection string on clipboard** button at the end of the property box.

4.4.1.2 Shared Access Signature (SAS)

Setting	Description
Endpoint suffix	Select the Microsoft cloud that hosts the storage account. Microsoft maintains separate clouds for data residency and regulatory compliance reasons.
Storage account name	All access to Azure Storage is done through a storage account. This account can be a general-purpose storage account or a Blob storage account specifically for storing objects/blobs.
Blob Container **Queue**	The top-level container name or the name of the queue. The property shown here differs depending on the **Type** (**Blob** or **Queue**) selected on the **Command** tab of the module's **Properties** window. This property is optional. If you do not supply the container or queue name here, you must supply it as a property of the command.
SAS Token	The SAS token generated for the storage account.

4.4.1.3 Application Registration

Setting	Description
Endpoint suffix	Select the Microsoft cloud that hosts the storage account. Microsoft maintains separate clouds for data residency and regulatory compliance reasons.
Storage account name	All access to Azure Storage is done through a storage account. This account can be a general-purpose storage account or a Blob storage account specifically for storing objects/blobs.
Tenant Domain	Click **…** (three dots) at the end of the value box to enter the tenant domain.
Client ID	The **Application (client) ID** of the application's Entra ID app registration. **Client ID** identifies the app.
Client Secret	A client secret generated for that app registration. **Client Secret** is essentially the application's password.

Important: This authentication method is not applicable to Azure Storage containers that are managed by Microsoft Dynamics 365 Finance and Operations.

Note: Implementing application registration authentication requires additional configuration in Entra ID (for the app registration) and for the specified storage account (to give that application identity permission to interact with the storage).

To do that configuration, an Azure administrator must follow the instructions in the following Lasernet knowledge base article: *Configure Microsoft Azure to Support Lasernet Access to Azure Storage Through App-Registration-Based Authentication*. As part of that process, the Azure administrator will supply you with the client ID and client secret that will enable you to configure application registration authentication for Azure Storage modules in Lasernet.

4.4.2 Command

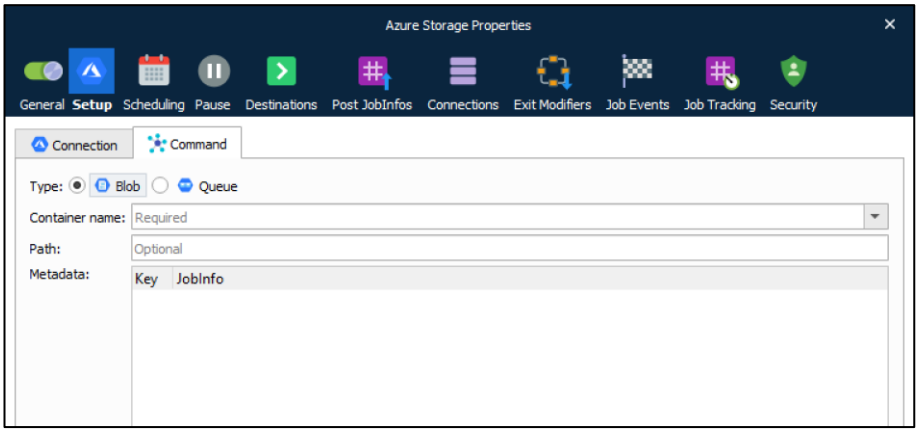
Select **Blob** to download from a container, or **Queue** to fetch messages.

Blob Files are downloaded from a specified Container.

Queue Messages are fetched from a specified Queue.

Please note that files and messages are deleted when polled.

4.4.2.1 Blob

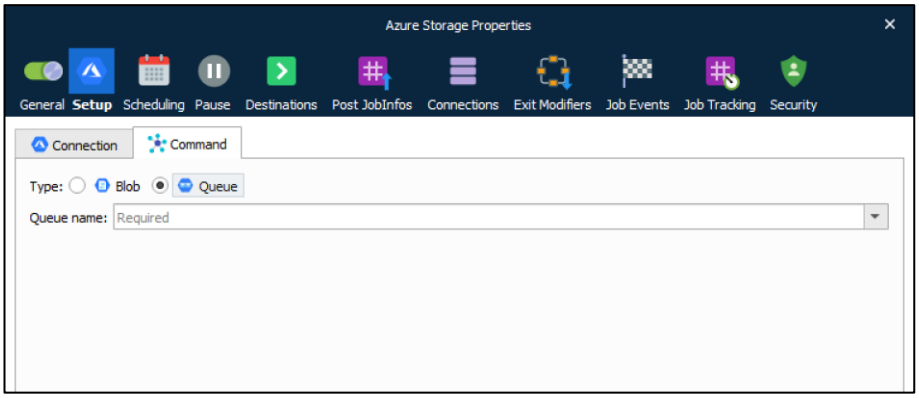


Container name (Required) The container where the BLOBs are located.

Path (Optional) The path within container.

Metadata Identify individual metadata fields (attached to the BLOB) that you want to be transferred to JobInfos.

4.4.2.2 Queue



Queue name (Required) The queue name.

4.4.3 JobInfos

The following JobInfos are set for Jobs created from messages:

AzureStorageMessageID GUID of the message.

AzureStorageMessagePopReceipt A BASE64 encoded GUID. The value of PopReceipt is opaque to the client; its only purpose is to ensure that a message may be deleted with the Delete Message operation.

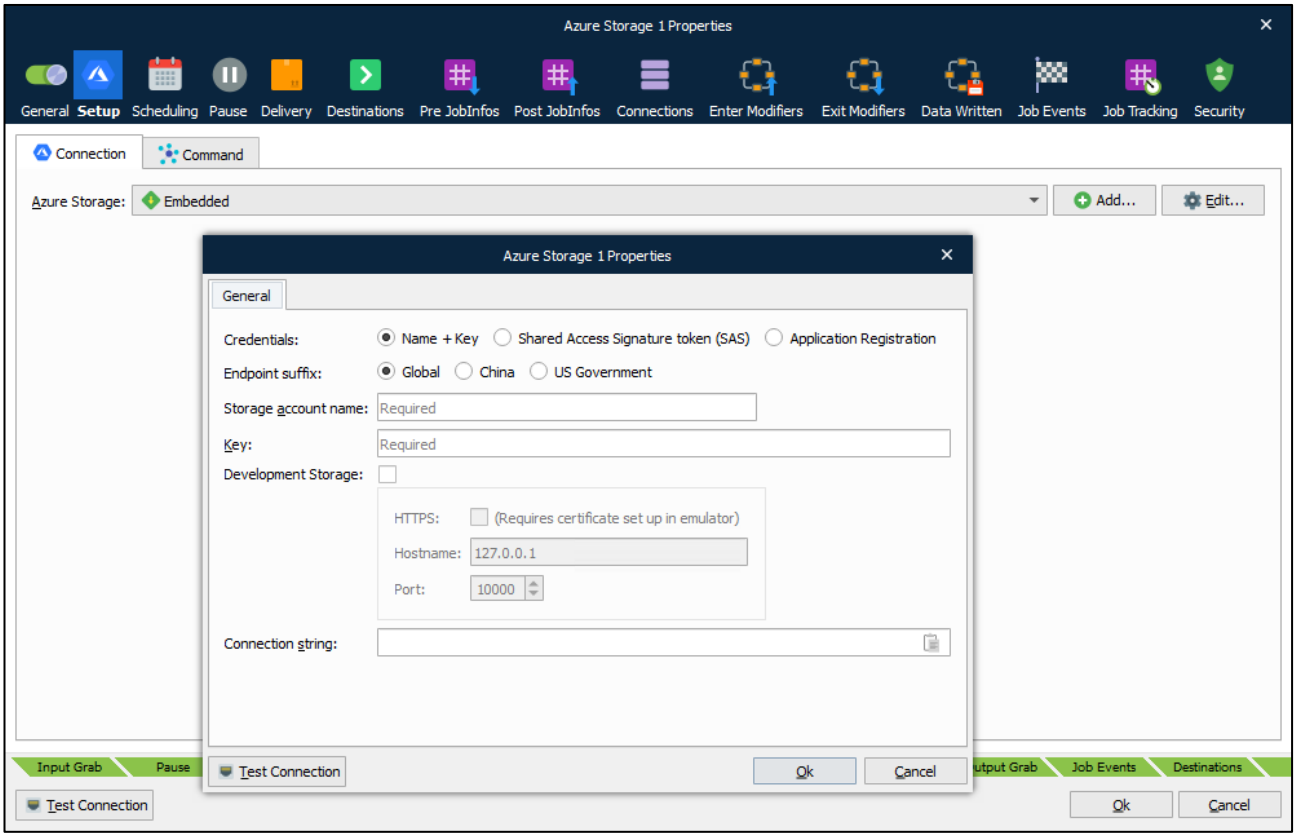
AzureStorageMessageDequeueCount The DequeueCount element has a value of 1 the first time the message is dequeued. This value is incremented each time the message is subsequently dequeued.

Extension The extension of the filename.

Filename	The name of the file without the path.
FilenameWithoutExt	The name of the file without the extension.
FullFilename	The name of the file including the path.
Filesize	The size of the file in bytes.
Mimetype	Content-type of the message.

4.5 Azure Storage Output Module

The Azure Storage output module is used to upload files to BLOB storage or to insert messages in Queues.



The setup of the module consists of two pages: **Connection** and **Command**.

4.5.1 Connection

In order to access the Azure Storage API, the module's **Connection** properties must be correctly configured.

This module's connection is configured in the same way as the Azure Storage Input Module.

See section 4.4.1 Connection in the Azure Storage Input Module section to learn about connection types (global and embedded) and how to configure each of the three available authentication options (name + key, Shared Access Signature token (SAS), and application registration).

4.5.2 Command

One of two types of commands are available:

Blob	Files are uploaded to a specified Container.
Queue	Messages are inserted in a specified Queue.

4.5.2.1 Blob

Container name	The container to upload the BLOBs to.
Path	The path (within the container) to upload to. This property is optional.
Blob name	The name of the file. This property is mandatory.
Metadata	Metadata fields that will be added to the BLOB in Azure Storage. To add a metadata field to the BLOB, click **Add** below the metadata table. In the **Add Metadata** window, enter a **Key** name for the metadata, then enter a value for the metadata in the **Value** box. You can select a JobInfo from the list (or enter a JobInfo name) to use dynamic data, or you can enter a static value.

4.5.2.2 Queue

Queue name	The queue to upload to.

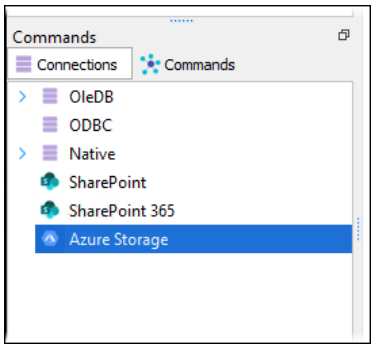
4.5.3 JobInfos

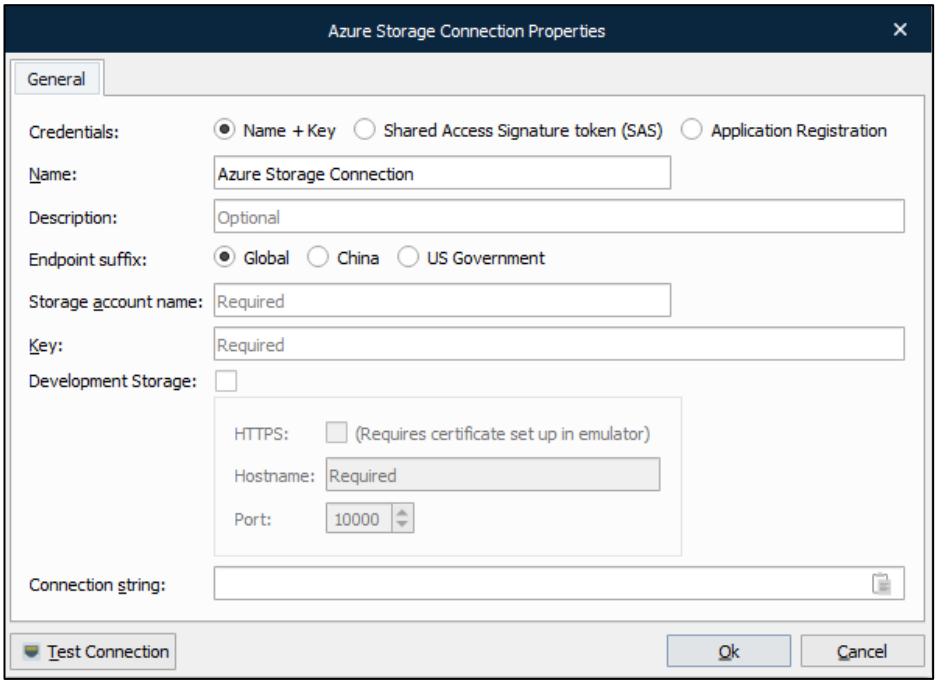
The following JobInfo is used by the Azure Storage output module:

Mimetype	Content-type of the message.

4.6 Azure Storage Connection

Connection to Azure Storage is added through the **Commands** menu in Lasernet.





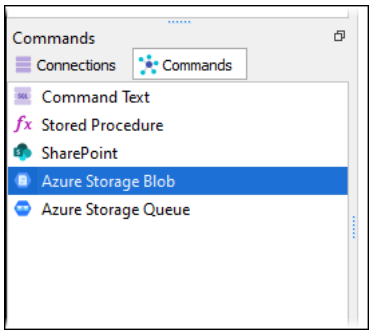
Connections added here have **Name** and **Description** properties to describe the connection itself.

Three types of authentication are available. For information on how to configure them, see:

- 4.4.1.1 Name + Key
- 4.4.1.2 Shared Access Signature (SAS)
- 4.4.1.3 Application Registration

4.7 Azure Storage BLOB Command

Azure Storage Blob commands are added through the **Commands** menu in Lasernet.

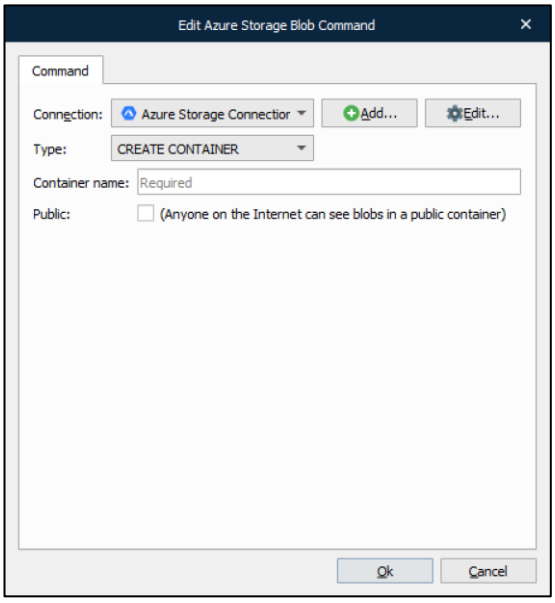


There are five different command types to choose from:

- Create Container
- Delete Container
- Upload Blob
- Download Blob
- Delete Blob

4.7.1 Create Container

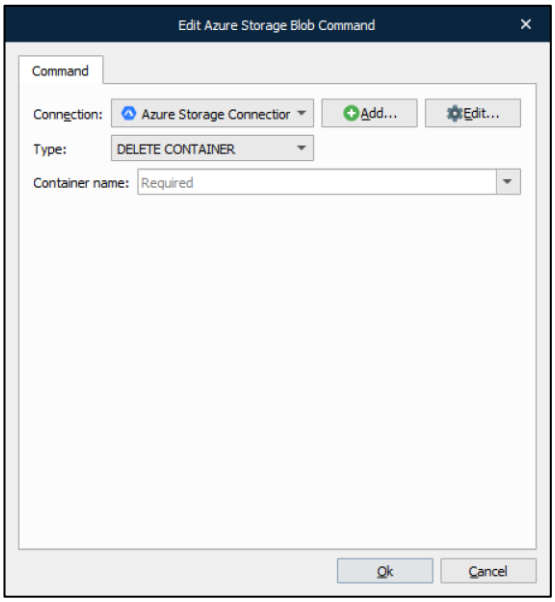
A container groups together of a set of blobs. All blobs must be in a container. An account can contain an unlimited number of containers. A container can store an unlimited number of blobs. *Please note: the container name must be lowercase*.



Anyone using the Internet can see blobs in a public container, but you can only modify or delete them if you have the appropriate access key.

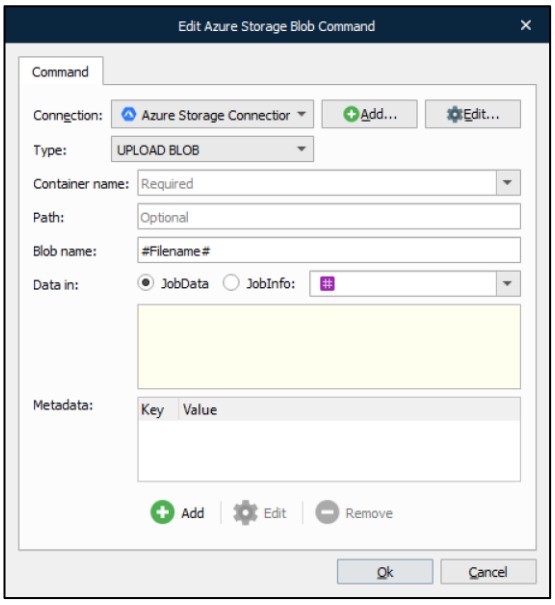
4.7.2 Delete Container

This command type is used to delete a container and all the blobs within in.



4.7.3 Upload Blob

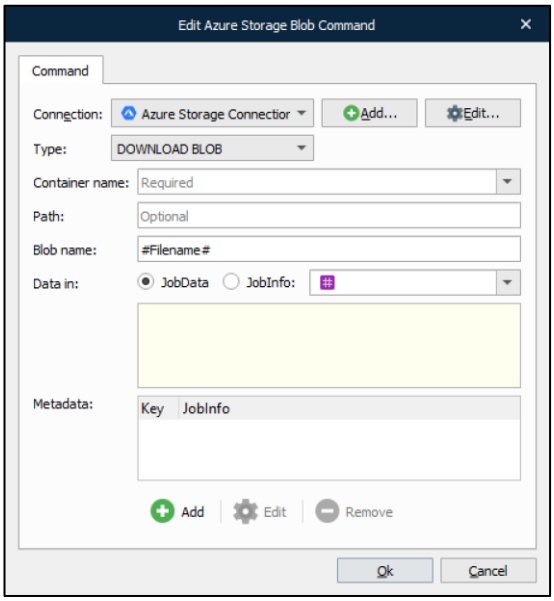
This command type is used to upload a file to a specified Container (using an optional path within the Container if required).



Data can be uploaded from either JobData or a JobInfo.

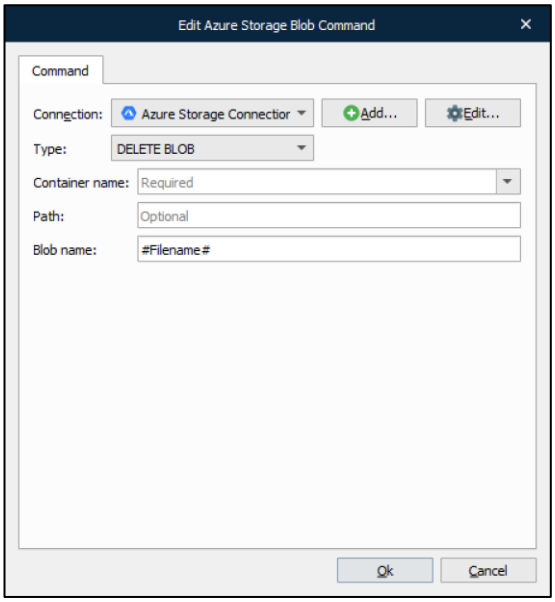
4.7.4 Download Blob

This command is used for downloading files by their given blob name.



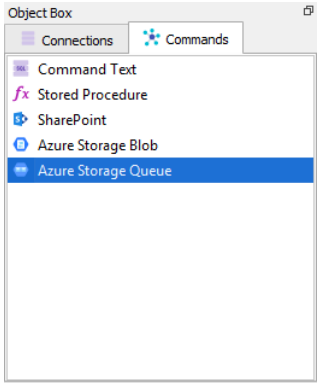
4.7.5 Delete Blob

This command type is used for deleting a blob by its given name.



4.8 Azure Storage Queue Command

Azure Storage Queue commands are added through the Databases menu in Lasernet.

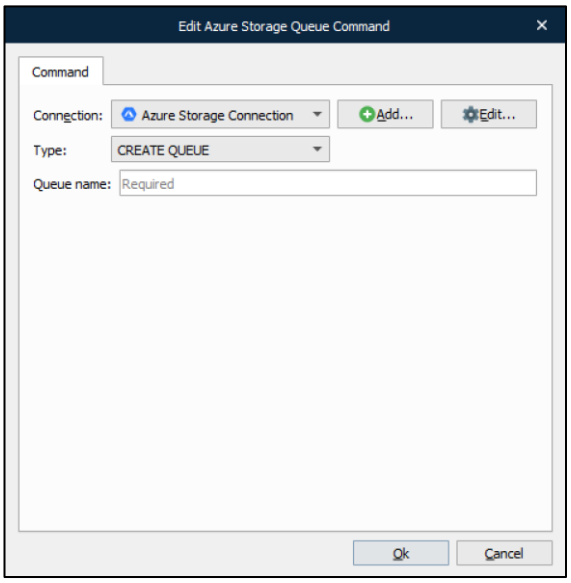


There are five different command types to choose from:

- Create Queue
- Delete Queue
- Insert Message
- Peek Message
- Get Message

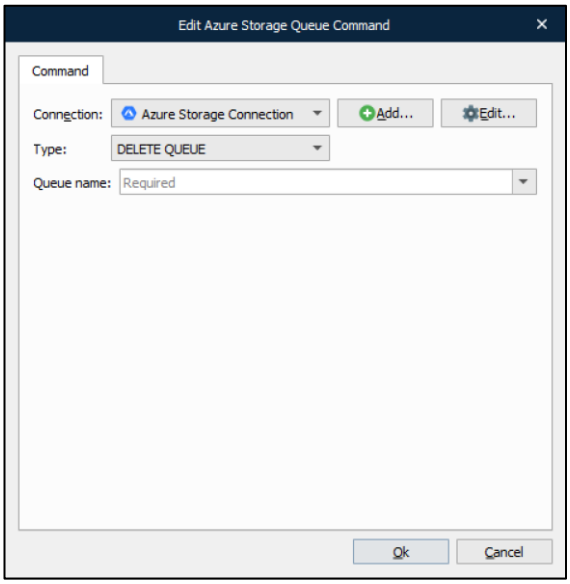
4.8.1 Create Queue

This command type is used for creating a message Queue.



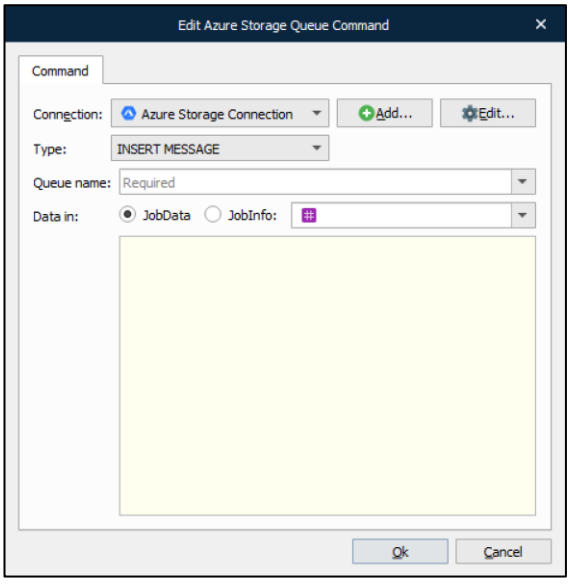
4.8.2 Delete Queue

This command type is used for deleting a message Queue along with all contained messages.



4.8.3 Insert Message

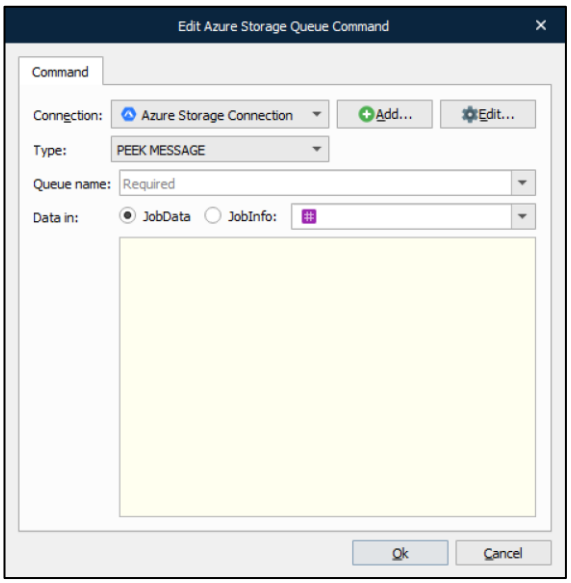
This command type is used for inserting a message in a specified Queue.



Data can be inserted from either JobData or a JobInfo.

4.8.4 Peek Message

This command type is used for peeking at a message in a specified Queue. Peeking a message does not delete it.

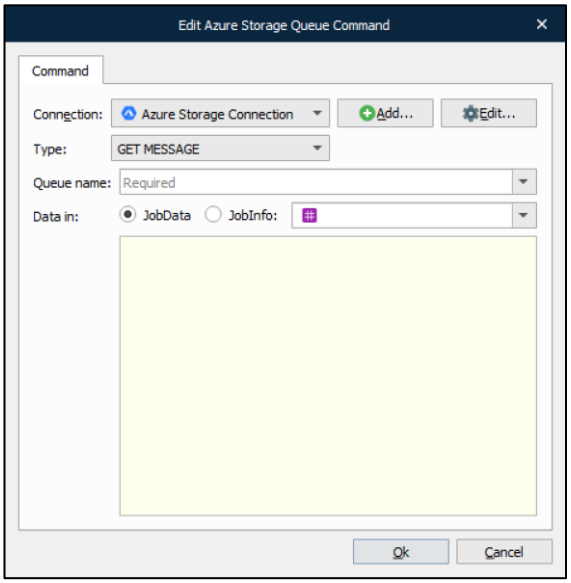


Data can either be inserted in JobData or a JobInfo.

For a list of JobInfos set, see Queue.

4.8.5 Get Message

This command type is used for fetching a message from a specified Queue. Fetching the message will delete it from the Queue.



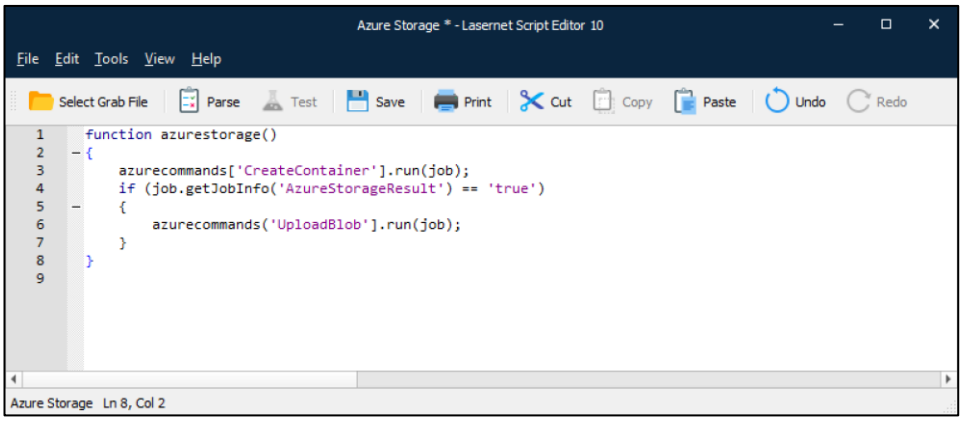
Data can either be inserted in JobData or a JobInfo.

For a list of JobInfos set, see Queue.

4.9 Azure Storage Scripting

It is possible to call Azure Storage commands directly via scripting in Lasernet, just like modifiers and SharePoint commands. This is done via an array called "**azurestorecommands**". In order to retrieve a specific command, use the index operator "[]" and a string with the name of the command as an argument. You can then use the "run" function to invoke the command.

Please see the screen-shot below for an example.



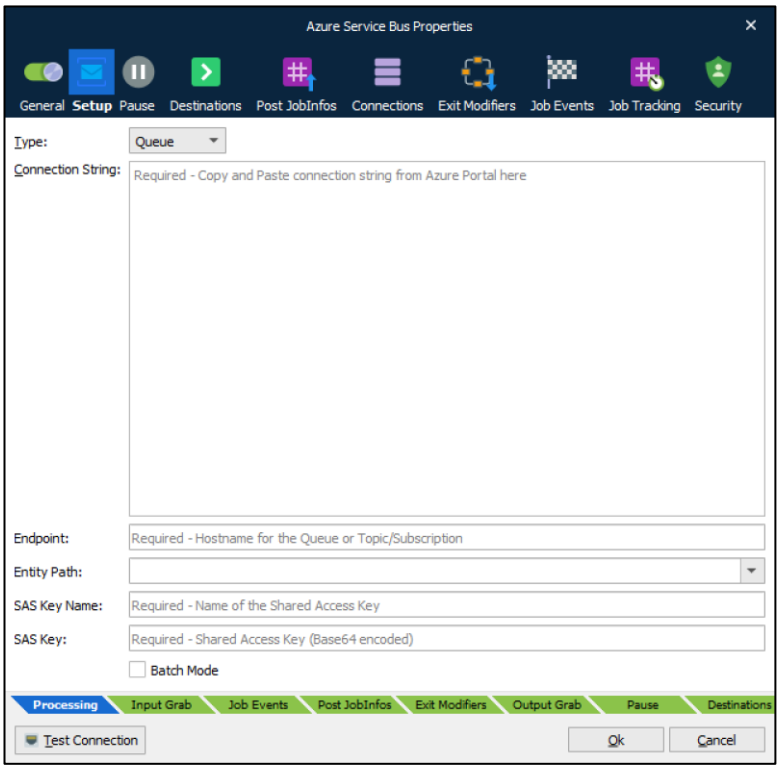
Tip: You can drag any Azure Storage Command from the configuration into the editor window and the script for running it (including the value and parameter), will be automatically inserted.

4.10 Azure Service Bus Input Module

The Azure Service Bus input module is used to pull messages from a Service Bus Queue or a Service Bus Subscription. Lasernet is notified of new messages by a Push-style notification system, so no Scheduler setup is required.

4.10.1 Configuration

To access the Azure Service Bus API, the properties must be correctly configured on the **Setup** page.



Type

Choose between **Queue** or **Topic**. Queues are used when you have one sender and one recipient, or if you need to balance the load by distributing messages between multiple recipients. If several input ports pull messages from the same queue, the messages will be distributed between them.

Topics are used when you have one sender and several recipients. Each recipient must have its own subscription. Messages are delivered once to each subscription.

Connection String

You can copy and paste the connection string from the Azure Portal. The remaining fields will be automatically populated by the content of the connection string. Alternatively, you can fill the remaining fields and the connection string will be automatically built.

Subscription

Only visible in Topic mode. Enter the name of the subscription to pull messages from.

Entity Path

The name/path of the Queue or Topic.

SAS Key Name

The name of the Shared Access Service (SAS) key.

SAS Key

The Shared Access Service (SAS) key used to authenticate Lasernet.

Batch Mode

If you select **Batch Mode**, the jobs that this input creates will run in batch mode.

Configure inputs that are used to receive larger, lower-priority jobs to use batch mode.

Configure inputs that are used to receive higher-priority jobs (such as processing previews for users) to not use batch mode, so that they run in standard multi-threaded mode and utilize available CPU cores.

Batch mode limits the proportion of CPU cores that are available to all jobs running in batch mode. This intentional limiting of CPU core usage ensures that sufficient cores remain available for other, higher-priority processing, while also enabling the multi-threaded, simultaneous processing of multiple batch mode jobs.

4.10.2 JobInfos

The following JobInfos are set for Jobs created from messages:

AzureServiceBusContentType	Content-type of message contents (see also MimeType).
AzureServiceBusCorrelationId	The CorrelationId for this message, if specified by sender.
AzureServiceBusDeliveryCount	The number of times the message has been delivered.
AzureServiceBusExpiresAtUtc	Date and time in UTC at which the message is set to expire (RFC1123 format.
AzureServiceBusForcePersistence	Sets a value to indicate whether the message is to be persisted to the database immediately instead of being held in memory for a short time. Ignored if message is sent to a non-express queue.
AzureServiceBusLabel	A sender-assigned Label for the message, if present.
AzureServiceBusLockedUntilUTC	Specifies when the initial lock expires – the lock is automatically extended by Lasernet if processing exceeds the lock timeout.
AzureServiceBusMessageId	The (sender-assigned) message Id. Can be used to detect duplicates.
AzureServiceBusMessagePropertyNames	Array of metadata field names to associate with the message.
AzureServiceBusMessagePropertyValues	Array of metadata field values to associate with the message.
AzureServiceBusPartitionKey	Partition key for sending a transactional message to a queue that is not session-aware.
AzureServiceBusReplyTo	The queue or topic to send replies to, if specified by sender.
AzureServiceBusReplyToSessionId	The session ID to send replies to on a session-aware queue, if specified by sender.
AzureServiceBusScheduledEnqueuedTimeUTC	The time when the message was submitted to the queue or topic.
AzureServiceBusSequenceNumber	The unique number assigned to the message.
AzureServiceBusSessionID	The SessionID for this message, if session-aware queue is used.
AzureServiceBusTimeToLive	The message's time to live value in seconds. This is the duration after which the message expires, starting from when the message is sent to the Service Bus.

AzureServiceBusTo	The send to address.
AzureServiceBusViaPartitionKey	Partition key value when a transaction is to be used to send messages via a transfer queue.
MimeType	The MIME type of the message content if specified by sender.

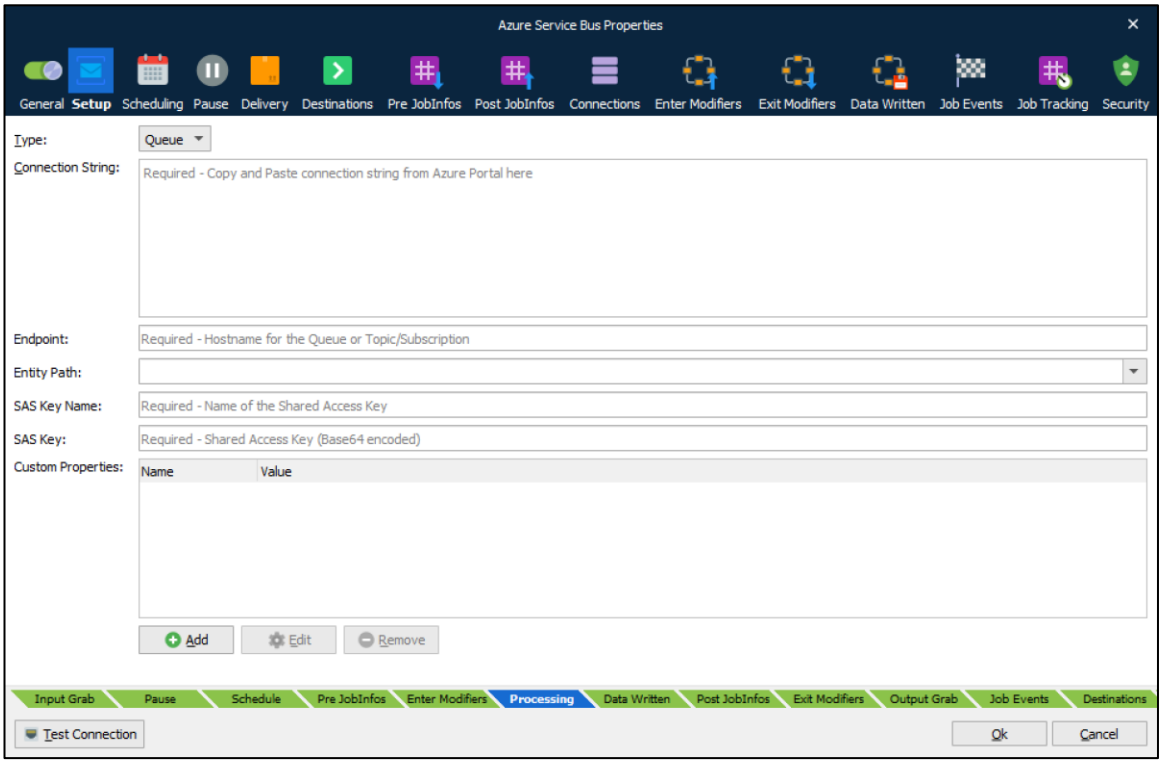
If any custom properties (metadata) are attached to the message, they will also be set as JobInfos. Properties are treated as key-value pairs. Objects and arrays (JSON) are not supported.

4.11 Azure Service Bus Output Module

The Azure Service Bus output module is used to push messages to either an Azure Service Bus Queue or an Azure Service Bus Topic. If messages are sent to a Topic, the message will be distributed to all Subscribers to that Topic.

4.11.1 Configuration

The Azure Service Bus output module requires several setting to be filled in order to gain access to the Azure Service Bus API.



Type	Choose between Queue or Topic. Queues are used when you have one sender and one recipient, or if you need to balance the load by distributing messages between multiple recipients. If several input ports pull messages from the same queue, the messages will be distributed between them. Topic is used when you have one sender and several recipients. Each

	recipient must have its own subscription. Messages are delivered once to each subscription.
Connection String	You can copy and paste the connection string from the Azure Portal. The remaining fields will automatically be populated by the content of the connection string. Alternatively, you can fill the remaining fields and the connection string will be automatically built.
Entity Path	This is the name/path of the Queue or Topic.
SAS Key Name	The name of the Shared Access Service (SAS) key.
SAS Key	The Shared Access Service (SAS) key used to authenticate Lasernet.

4.11.2 Custom Properties / Metadata

It is possible to add custom properties to a message sent via the Azure Service Bus. Custom properties are key-value pairs which are distributed alongside the message. The custom properties are sent as HTTP headers. This imposes some restrictions on the content of the key. It must abide by the restrictions for header names defined in RFC2616.

If you use characters that are not allowed according to RFC2616, Lasernet will either remove them (ASCII 128 or higher), or replace them with hyphens (control-characters and separator-characters).

The value will be encoded as a JSON string in the HTTP request to the Azure Service Bus API.

4.11.3 JobInfos

The following JobInfos are used by the Azure Service Bus output module:

AzureServiceBusCorrelationId	The CorrelationId for this message.
AzureServiceBusForcePersistence	Force the message to be written to persistent storage immediately.
AzureServiceBusLabel	A sender-defined label.
AzureServiceBusMessageID	Identifier of the message. This is a user-defined value that the Service Bus can use to identify duplicate messages.
AzureServiceBusMessagePropertyNames	Array of metadata field names to associate with the message (see section 4.8.4).
AzureServiceBusMessagePropertyValues	Array of metadata field values to associate with the message (see section 4.8.4).
AzureServiceBusMimeType	The MIME type of the message content. Defaults to application/octet-stream if not set.
AzureServiceBusPartitionKey	A partition key for sending a transactional message to a queue or topic that is not session-aware.
AzureServiceBusReplyTo	The address of the queue to send replies to.
AzureServiceBusReplyToSessionId	The SessionId to reply to.
AzureServiceBusScheduledEnqueueTimeUtc	The date and time in UTC at which the message will be enqueued. Specified as a RFC2822 date/time string. Example: “Thu, 01 Sep 2016 08:28:01 GMT”.
AzureServiceBusSessionId	The identifier of the session.
AzureServiceBusTimeToLive	The message’s time to live in seconds. Messages older than the time-to-live value will expire and no longer be retained in the message store.
AzureServiceBusTo	The send-to address.

AzureServiceBusViaPartitionKey	Sets a partition key value when a transaction is to be used to send messages via the transfer queue.

Each of these JobInfos (except MimeType) corresponds to a similar named property of a Service Bus BrokeredMessage object. For more information about these properties, please refer to Microsoft's documentation of the Azure Service Bus and Brokered Messaging.

4.11.4 Custom Properties / Metadata using JobInfos

If the custom properties of the message are not static, it is possible to add dynamic custom properties to a message by using JobInfos. As described in 4.11.2, custom properties are key-value pairs. Therefore, you must manipulate two JobInfos to add a custom property;

AzureServiceBusMessagePropertyNames	The name of the key for the custom properties. See also 4.11.2 about character restrictions of message property keys.
AzureServiceBusMessagePropertyValues	The value of the custom properties.

The JobInfos are treated as arrays, where the first entry in MessagePropertyNames defines the key of the first property, and the first entry of the MessagePropertyValues defines the value of the first property, and so forth.