

User guide

[Edit](#)[New Page](#)[Jump to bottom](#)

Ahmed Elharouny edited this page 3 days ago · 49 revisions

Setup and permissions

Setup for your tenant

To setup Service Bus Cloud Explorer to be used for your tenant; create a `Service Bus Cloud Explorer` resource in Azure Portal or from [Azure Marketplace](#). For more details on setup process; please visit: [Setup in your tenant](#) page.

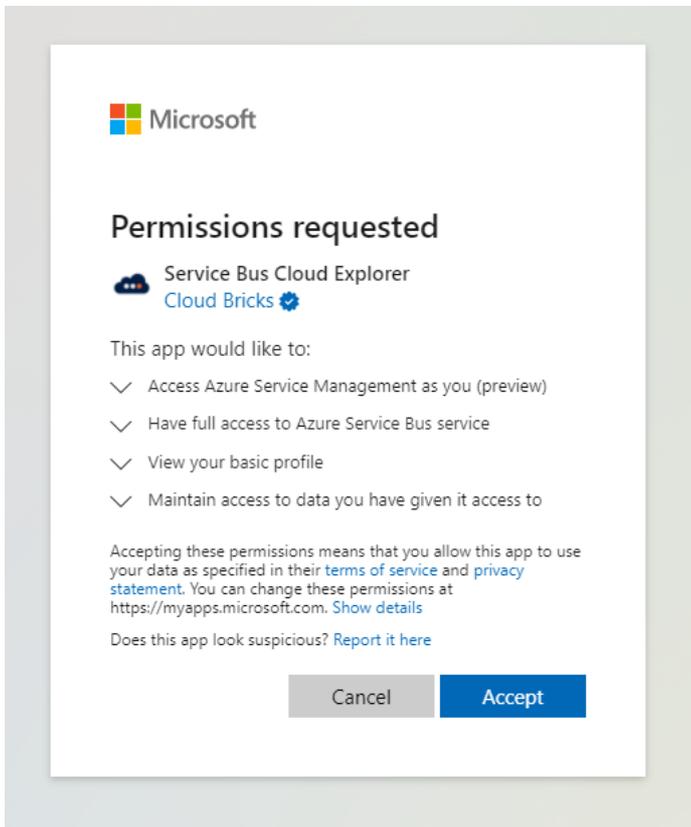
Start the app and login

To start Service Bus Cloud Explorer; users need to navigate to servicebus.cloudbricks.io and login with their organization accounts.

Provide Consent

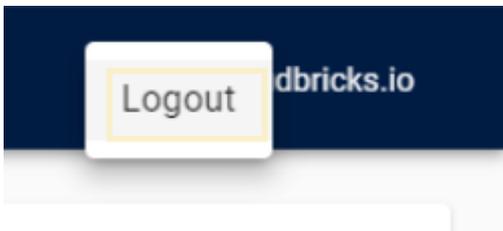
During the first time any user logs in to the app, a consent screen will show up (if your organization administrator didn't consent on behalf of all users). Users have to consent on the following:

1. **Access Azure Service Management As you.** This is required as the app users Azure Service Management APIs to load all namespaces, topics, queues and other information like active message counts and dead letter queue counts.
2. **Have full access to Azure Service Bus service** This is required to be able to perform service bus operation on behalf of logged in user like peeking a message off a queue or sending a message to a topic. (**Please note** that App access is limited to the access of the user despite what appears in the title.)
3. **View your basic profile.** This is required to log in users and get access to basic information like user email.
4. **Maintain access to data you have give it access to.** This is required to be able to use refresh tokens to avoid having to log in users every time they use the app.



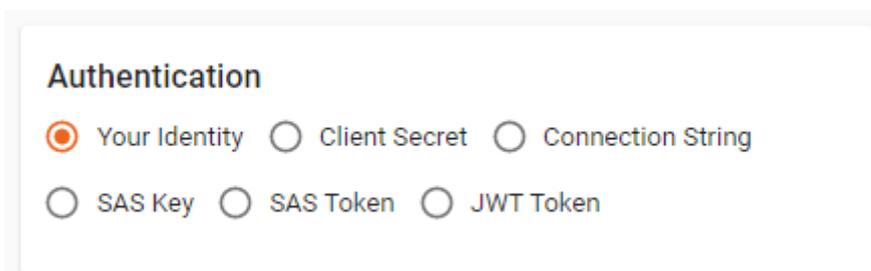
Log out

The name of the logged-in user will appear in the application top bar. To log out; click on user name and click `log out` button from the popup menu.



Authenticate via user identity

The default method to authenticate to Service Bus entities is via user's Azure AD identity.



A basic read access over Service Bus namespaces is enough to populate all entities in left sidebar. In addition to that; the following permissions might be needed:

- To send messages via user identity; users need `Azure Service Bus Data Sender` permission.

- To peek or receive messages via user identity; users need Azure Service Bus Data Receiver permission.

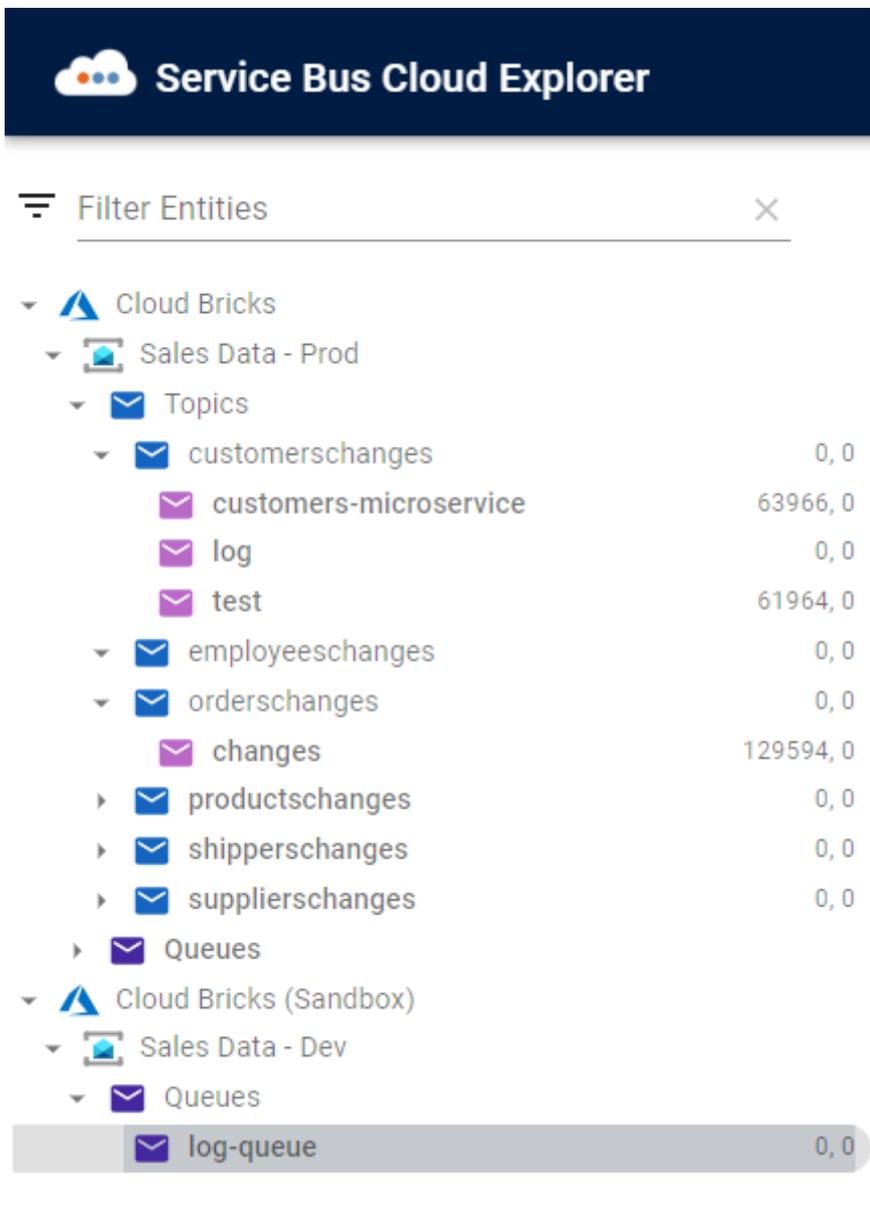
The following screenshot is showing the permissions applied on a Resource Group level:

The screenshot shows the Azure portal interface for a resource group named 'ProofOfConcepts'. The 'Access control (IAM)' section is active, and the 'Role assignments' tab is selected. The interface displays the number of role assignments for the subscription (8 out of 2000) and a search bar. A table lists the role assignments, with three items shown:

Name	Type	Role	Scope	Condition
▼ Azure Service Bus Data Receiver				
<input type="checkbox"/>	User	Azure Service Bus Data Receiver	This resource	None
▼ Azure Service Bus Data Sender				
<input type="checkbox"/>	User	Azure Service Bus Data Sender	This resource	None
▼ Reader				
<input type="checkbox"/>	User	Reader	This resource	None

Explore Service Bus Entities

On the left sidebar, a tree of all namespaces in which the logged-in user have access to will show up. Users can expand nodes of the tree, which will load selected entities i.e. topics, queues and subscriptions on demand. Multiple namespaces can be expanded at the same time:



The screenshot shows the Service Bus Cloud Explorer interface. At the top, there is a dark blue header with the Service Bus logo and the text "Service Bus Cloud Explorer". Below the header, there is a "Filter Entities" search bar with a close button (X). The main content area displays a tree view of entities. The tree is filtered to show only entities containing "log" in their name. The entities are organized into two main sections: "Cloud Bricks" and "Cloud Bricks (Sandbox)".

Entity Name	Count
Cloud Bricks	
Sales Data - Prod	
Topics	
customerschanges	0, 0
customers-microservice	63966, 0
log	0, 0
test	61964, 0
employeeschanges	0, 0
orderschanges	0, 0
changes	129594, 0
productschanges	0, 0
shipperschanges	0, 0
supplierschanges	0, 0
Queues	
Cloud Bricks (Sandbox)	
Sales Data - Dev	
Queues	
log-queue	0, 0

Filter entities

Users can filter topic and queues by name to narrow down the tree to a subset or a specific entity.

The screenshot shows the Service Bus Cloud Explorer interface. At the top, there is a dark blue header with the Service Bus logo and the text "Service Bus Cloud Explorer". Below the header, there is a "Filter Entities" search bar containing the text "log-queue". The main area displays a tree view of entities:

- Cloud Bricks
 - Sales Data - Prod
 - Topics
 - Queues
 - log-queue 75865, 1
 - Cloud Bricks (Sandbox)
 - Sales Data - Dev
 - Queues
 - log-queue 0, 0

Message counts

In left sidebar you can see active and dead-lettered message counts for all entities in the format: [active], [dead-lettered].

The screenshot shows the left sidebar of the Service Bus Cloud Explorer interface. The tree view is expanded to show the following entities and their message counts:

Cloud Bricks	
Sales Data - Prod	
Topics	
customerschanges	0, 0
customers-microservice	63966, 0
log	0, 0
test	61964, 0
employeeschanges	0, 0
orderschanges	0, 0
changes	129594, 0
productschanges	0, 0
shipperschanges	0, 0
supplierschanges	0, 0

Select entities

Users can set selected entity by clicking a topic, queue or subscription on the left sidebar. Which will reflect on entity information window:

Service Bus Cloud Explorer

Filter Entities: log-queue

Cloud Bricks

- Sales Data - Prod
 - Topics
 - Queues
 - log-queue 75865, 1
- Cloud Bricks (Sandbox)
 - Sales Data - Dev
 - Queues
 - log-queue 0, 0

BROWSE SEND PEEK RECEIVE REPLAY

Dead Letter Queue Limit: 10,000

S...	Body	Enqueued Time	Message Id	D...
[Empty]	14	15/04/2022, 14:39:56	2fdd3cd551da4c77bf9f62...	2
[Empty]	15	15/04/2022, 14:39:59	a18d7cf44a494b40b3662...	1
[Empty]	16	15/04/2022, 14:40:02	8b02dc277f994c8093592...	1
[Empty]	17	15/04/2022, 14:40:06	e0499d1ab69a46d0ad479...	1
[Empty]	18	15/04/2022, 14:40:11	b3fa672f3fa45eea10922...	1
[Empty]	19	15/04/2022, 14:40:16	714ad24b71fa4fea2d2c6...	1

Entity Information

Namespace: https://cloudbricks-sales-data.servicebus.windows.net:443/
 Name: log-queue
 Type: queue

Active: 75865 messages Dead-Lettered: 1 messages
 Scheduled: 0 messages

Authentication

Your Identity Client Secret
 Connection String SAS Key
 SAS Token JWT Token

Access Service Bus Messages

In the middle of the screen, the tab navigation is showing all the actions that can be performed on Service Bus messages: Browse , Send , Peek , Receive and Replay .

Browse Messages

Browse messages enables users to browse all messages in queues and subscriptions without locking them (i.e. message retry count will not be increased upon reading messages). For more information check [message-browsing](#).

BROWSE SEND PEEK RECEIVE REPLAY

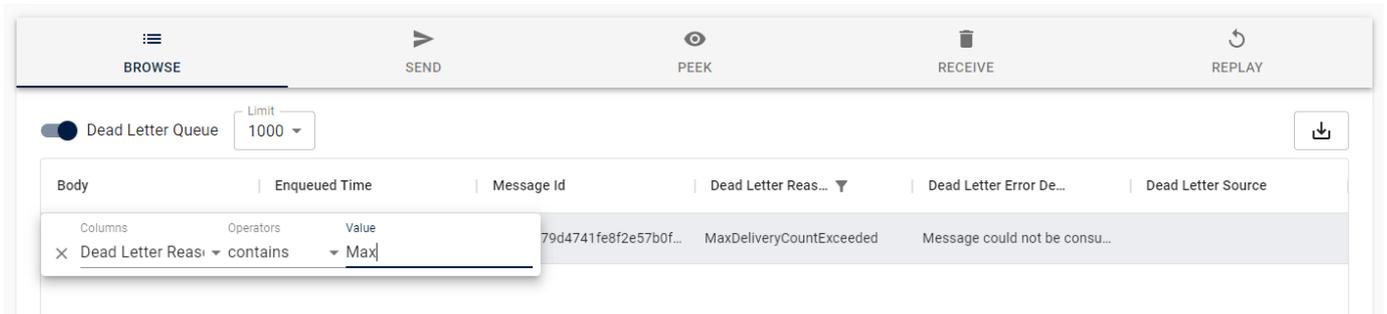
Dead Letter Queue Limit: 1000

Subject	Body	Enqueued Time	Message Id	Delivery Co...
[Empty]	[Empty]	16/04/2022, 17:23:00	9ee846d898a14564bac0b062c29e0dd4	1
[Empty]	A software engineer at Aligent Consulting with ...	17/04/2022, 00:42:29	e39adc731efa4ef18d852b4e1ca85981	1
Some subject	[Empty]	17/04/2022, 15:22:13	msg id	1
[Empty]	Message test with custom properties	18/04/2022, 20:29:22	bfa94b5993a4c7ebc7294d278cb95fc	1
[Empty]	Message test with custom properties 2	18/04/2022, 20:34:56	380050ed4bf74a99966653038fd5a025	1
[Empty]	test body	19/04/2022, 08:48:25	05989d4f7abc4404b9afa7f0abeaf46e	1
[Empty]	test body	19/04/2022, 08:48:25	1486b4fee33448d9a050e3e8bc521010	1
[Empty]	test body	19/04/2022, 08:48:25	051834b7fde0452f8d1f8d64ebb3f876	1
[Empty]	test body	19/04/2022, 08:48:25	036b9a78965e4dcd932f46532a23bdb8	1

Rows per page: 100 1-100 of 4500

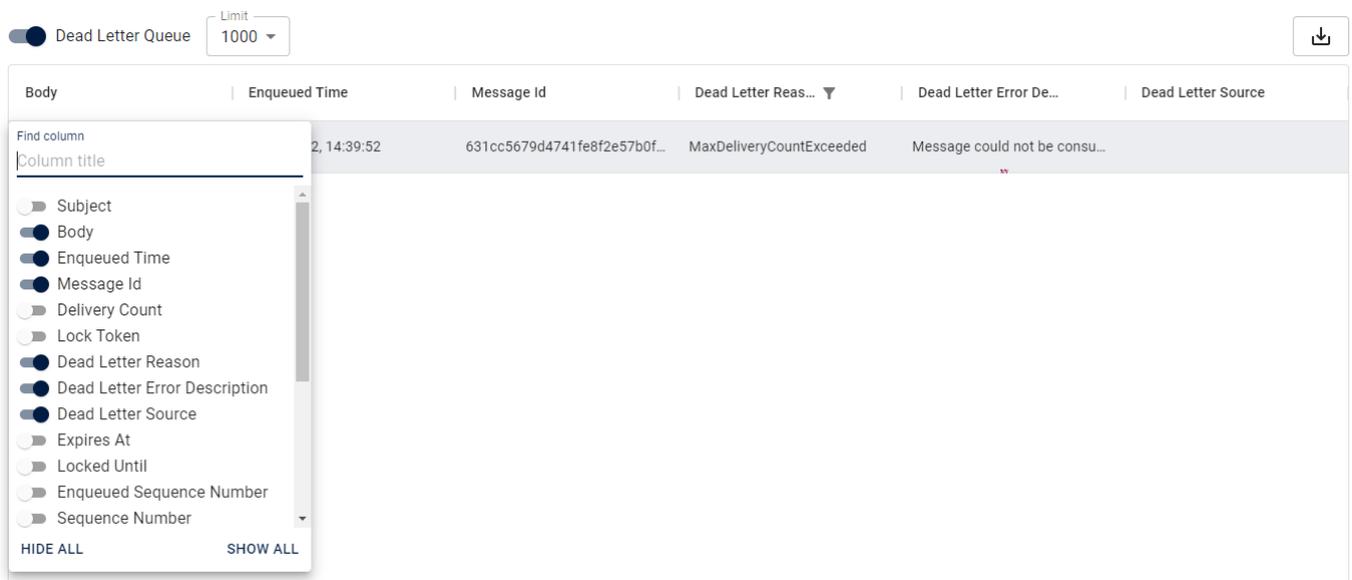
Filtering

Use the filtering functionality to filter rows of the grid via a particular criteria.



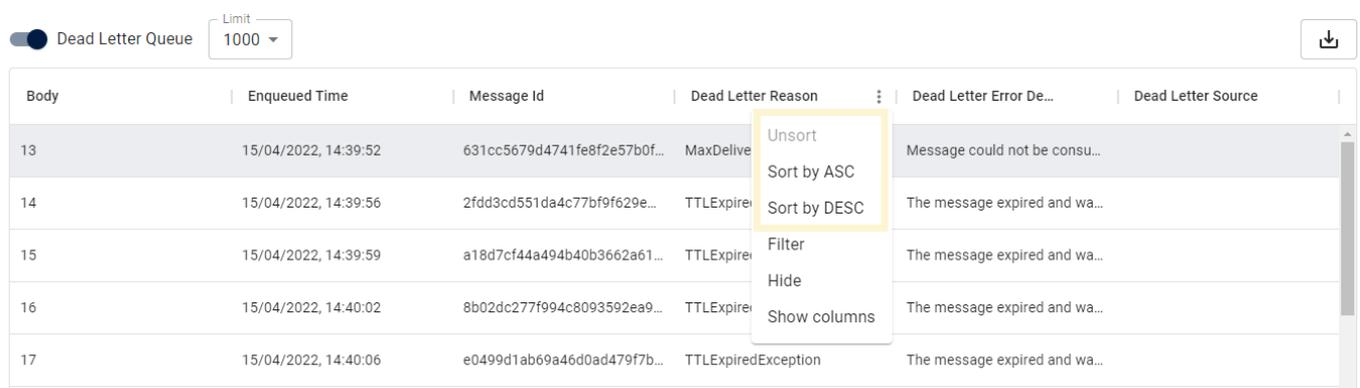
Show/Hide columns

Browse messages grid by-default displays few of the most important columns. However, you can use the show/hide column features to control what should be displayed.



Ordering

Use the ordering functionality to order by any column ascending or descending.

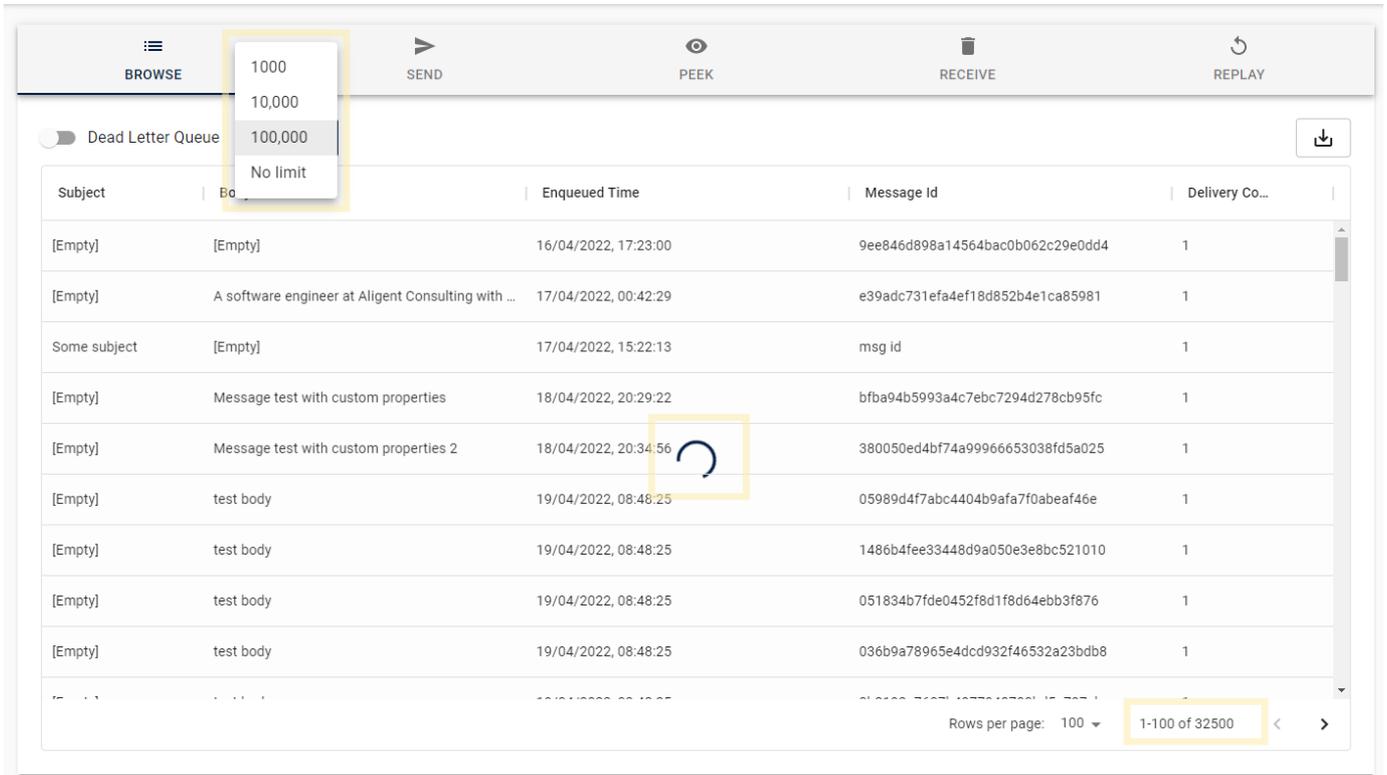


Apply limit

If you're dealing with a large number of messages you can use the limit dropdown to define an approximate number of messages to retrieve to save time and resources.

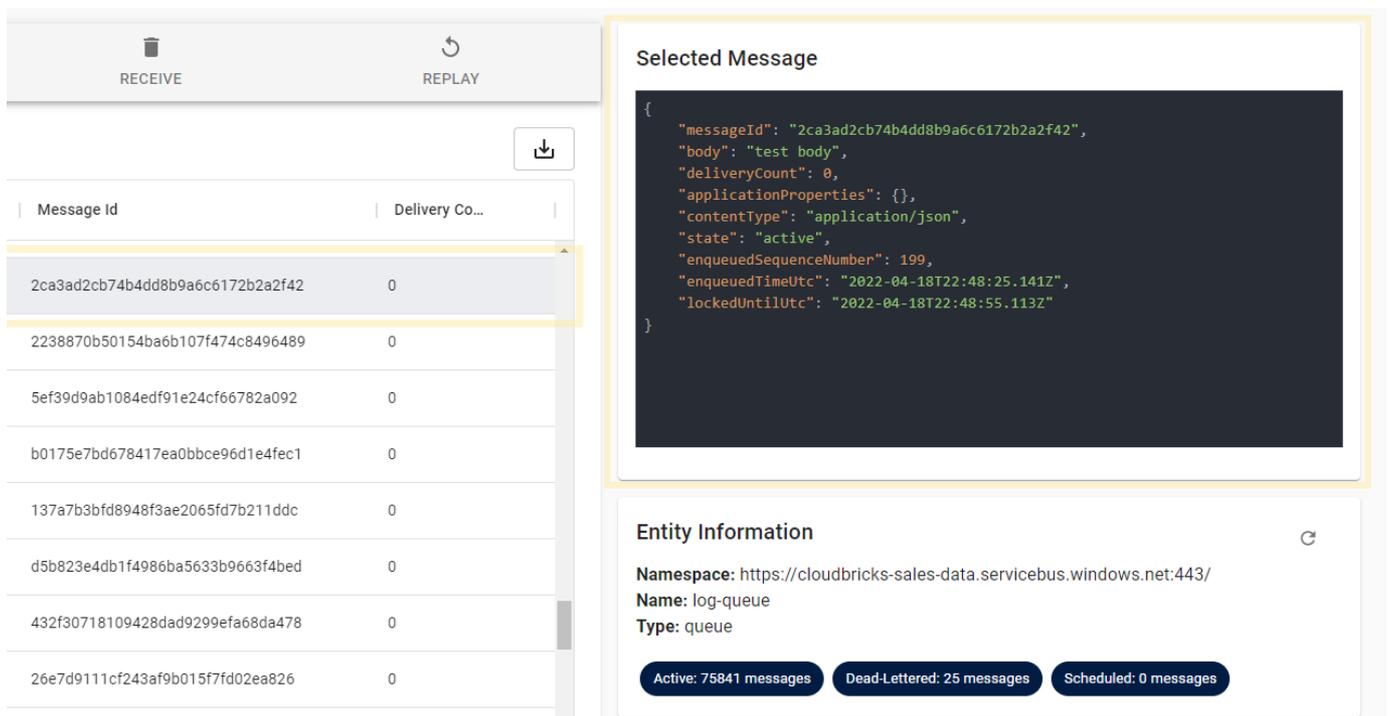
- Loading messages will pause if number of loaded messages equals or more than defined limit.
- Loading messages will resume if number of loaded messages are less than limit.

In the paging section you can see exactly how many messages loaded in memory.



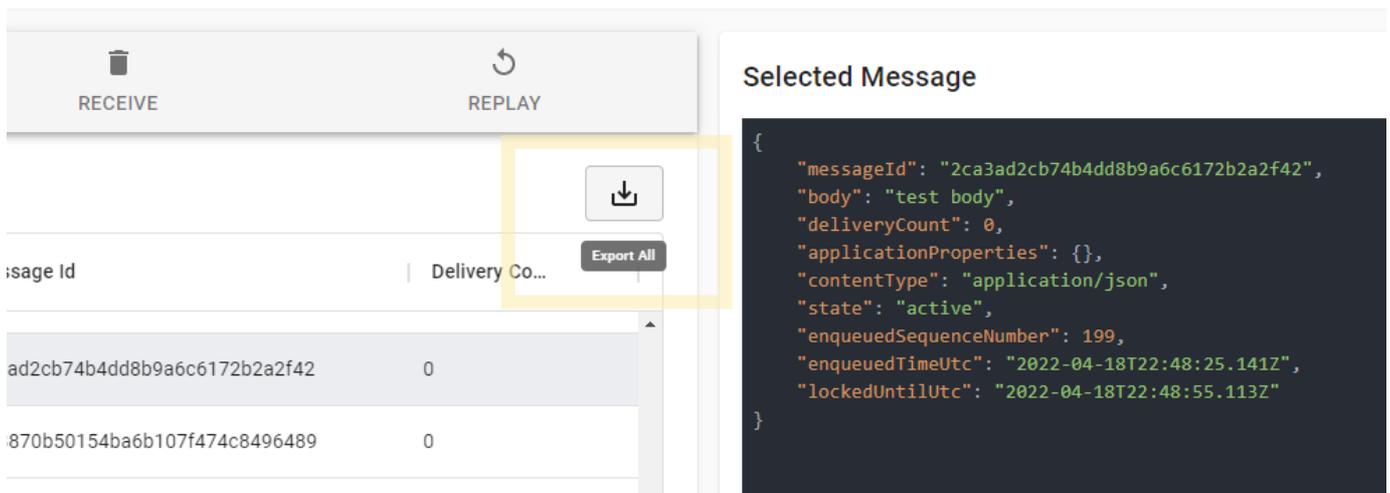
Show Message Details

Click on any row on the grid to select the message and reveal its full details in the right sidebar. Press **ctrl** and click a selected row to unselect.



Export Messages

Click `Export all` button to export all messages currently loaded in memory in JSON format.



The screenshot shows the Service Bus Cloud Explorer interface. At the top, there are 'RECEIVE' and 'REPLAY' buttons. Below them is a table of messages with columns for 'Message Id' and 'Delivery Co...'. The 'Export All' button is highlighted with a yellow box. To the right, the 'Selected Message' panel displays a JSON object representing a message.

```

{
  "messageId": "2ca3ad2cb74b4dd8b9a6c6172b2a2f42",
  "body": "test body",
  "deliveryCount": 0,
  "applicationProperties": {},
  "contentType": "application/json",
  "state": "active",
  "enqueuedSequenceNumber": 199,
  "enqueuedTimeUtc": "2022-04-18T22:48:25.141Z",
  "lockedUntilUtc": "2022-04-18T22:48:55.113Z"
}

```

Export functionality exports a JSON array of objects where each object represents a message where message schema is defined [here](#).

The following example is for an exported JSON document:

```

[
  {
    "messageId": "631cc5679d4741fe8f2e57b0fea03b81",
    "body": {
      "foo": "bar"
    },
    "timeToLive": 1209600000,
    "deliveryCount": 10,
    "applicationProperties": {
      "DeadLetterReason": "MaxDeliveryCountExceeded",
      "DeadLetterErrorDescription": "Message could not be consumed after 10 delivery"
    },
    "contentType": "application/json",
    "state": "active",
    "enqueuedTimeUtc": "2022-04-15T04:39:52.127Z",
    "lockedUntilUtc": "2022-04-26T06:17:23.204Z",
    "deadLetterReason": "MaxDeliveryCountExceeded",
    "deadLetterErrorDescription": "Message could not be consumed after 10 delivery at",
    "expiresAtUtc": "2022-04-29T04:39:52.127Z"
  }
]

```

For documentation on each of those fields, please check relevant properties in [Message Interface](#)

Send Messages

To send messages switch to `send` tab. Sending messages requires sender permission on the target entity.

The screenshot shows the 'SEND' tab interface. At the top, there are navigation buttons: BROWSE, SEND (active), PEEK, RECEIVE, and REPLAY. Below the navigation bar, there is a 'Content Type' field with the value 'application/json'. Underneath is a 'Body' field containing a JSON object: `{ "foo": "bar" }`. The 'Advanced Properties' section contains several input fields: Correlation ID, Message ID, Partition key, Session ID, Subject, To, Reply To, and Reply To Session ID. There are also two toggle switches: 'Time To Live' and 'Schedule Message'. Below this is the 'Custom Properties' section, which has two rows of 'Name' and 'Value' fields. The first row has 'custom' as the name and 'value1' as the value. At the bottom, there is a 'SEND' button and an 'Import mode' toggle switch.

For documentation on each of those fields, please check relevant properties in [Message Interface](#)

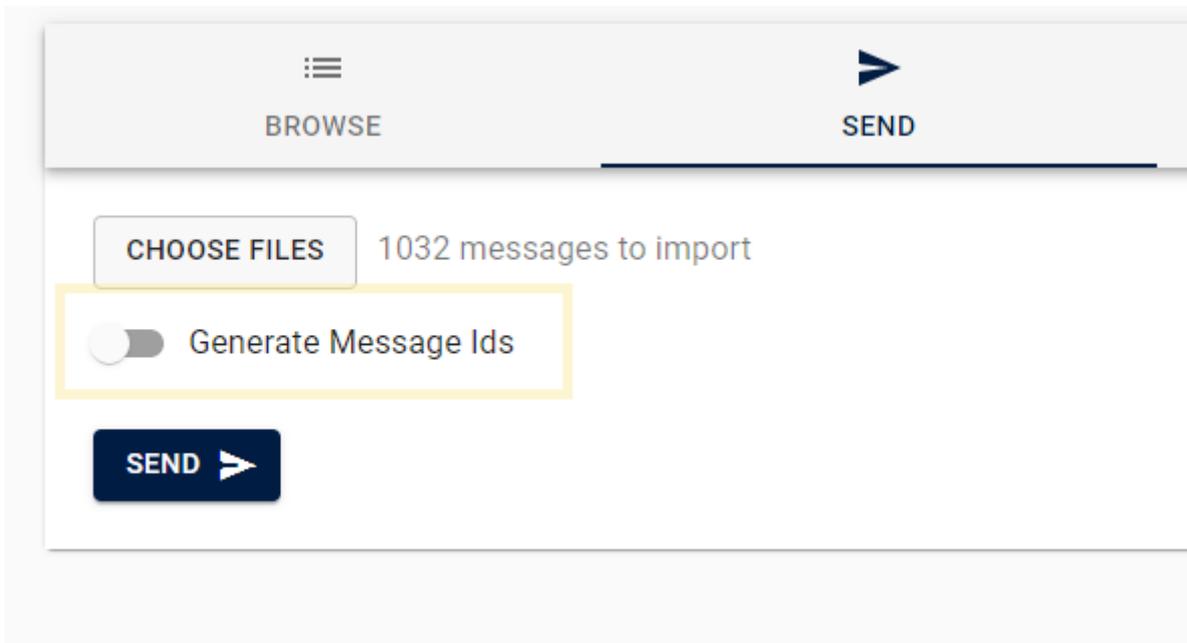
Import Messages

To send multiple messages you can switch to `Import mode` where you can upload a JSON file with an array of messages following schema defined [here](#).

The screenshot shows the 'SEND' tab interface in 'Import mode'. The 'CHOOSE FILES' button is highlighted with a yellow box and shows '1032 messages to import'. Below this is a 'Generate Message Ids' toggle switch, which is currently turned on. At the bottom, there is a 'SEND' button and an 'Import mode' toggle switch, which is also turned on.

messageId handling during import

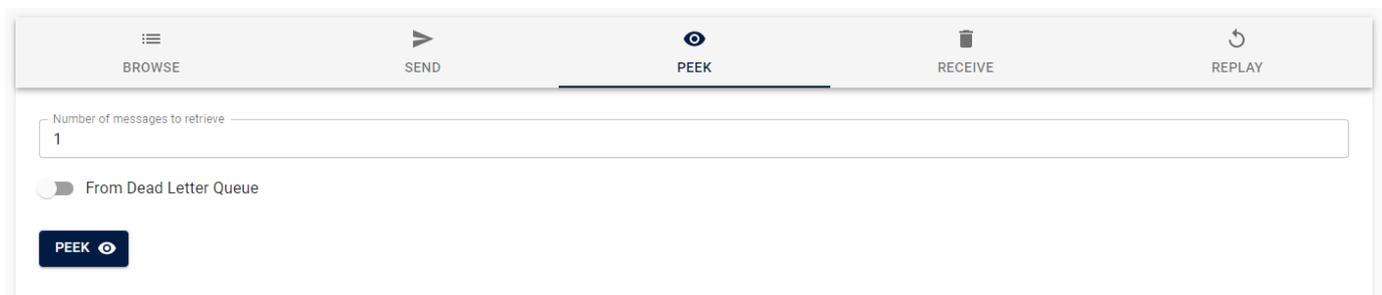
By default even if JSON file includes messages with `messageId` property; the app removes those properties to avoid the chance of having messages with duplicate `messageId` properties. To change that behaviour turn off `Generate Message Ids` option.



Peek Messages

Peeking is a *Non-destructive Read* operation of messages. To peek a single or multiple messages; switch to `peek` tab, select number of messages to peek and click `peek` button.

Note this uses the [Peek-Lock message operation](#). After reading the message the lock is released via an `abandon` operation which will increase `deliveryCount` and potentially could move the message to `dead-letter` queue if max retry count reached. This could be useful in some testing scenarios, however, if you want to read messages without locking please use [Browse Messages](#) instead.



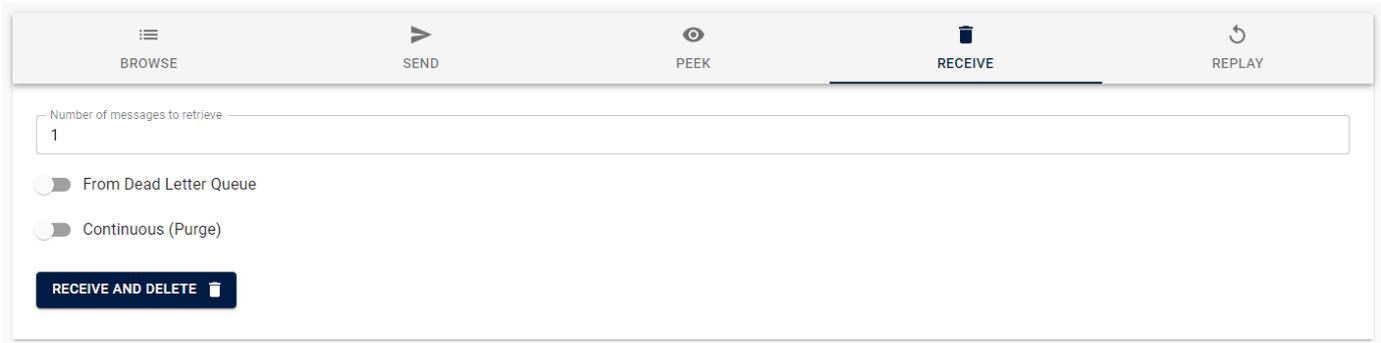
Peeked messages will show up in `Event Log` view. Click on status or message URLs to see all the details of loaded messages and the entire operation.

Time	Namespace	Entity	Type	Operation	Message	Status
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{"name":"ABC"}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{"name":"XYZ"}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{"name":"Fog"}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{"name":"Bar"}	Succeeded

Rows per page: 25 1-4 of 4

Receive Messages

Receiving is a *Destructive Read* operation of messages and will permanently delete the message(s) from Service Bus. To receive/delete a single or multiple messages; switch to receive tab, select number of messages to receive and click receive button.



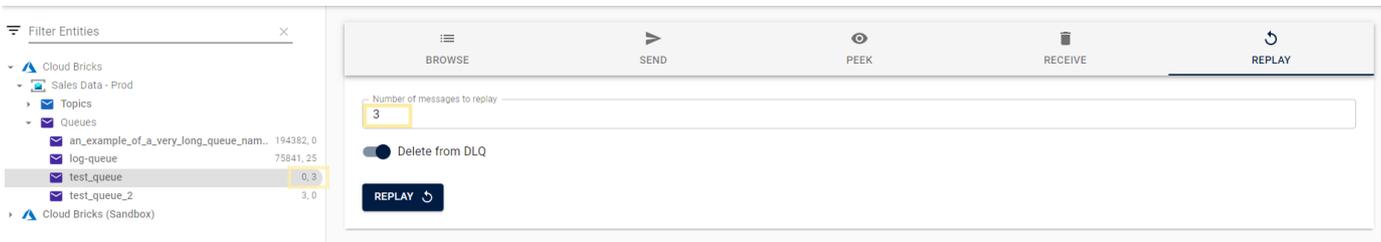
Received/deleted messages will show up in Event Log view. Click on status or message URLs to see all the details of those messages and the entire operation.

Time	Namespace	Entity	Type	Operation	Message	Status
10:59:07	cloudbricks-sales-data	test_queue_2	queue	receiveAndDelete	{'name':'Foo'}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{'name':'ABC'}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{'name':'XYZ'}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{'name':'Foo'}	Succeeded
10:51:09	cloudbricks-sales-data	test_queue_2	queue	peekLock	{'name':'Bar'}	Succeeded

Replay Messages

Replay functionality can be used to resubmit messages ended up in dead-letter queues and submitting them to either queues or topics and optionally delete messages from dead-letter queues after being resubmitted.

Note Replaying messages in a subscription's dead-letter queue will resubmit messages to subscription's topic. If topic has multiple subscriptions this may -based on filters- end up submitting messages into other subscriptions.



All replay operations will be show up in Event Log table.

Time	Namespace	Entity	Type	Operation	Message	Status
11:14:54	cloudbricks-sales-data	customerschanges/test/\$deadLetterQueue	subscription	complete	test_body	Succeeded
11:14:54	cloudbricks-sales-data	customerschanges	topic	send	test_body	Succeeded
11:14:54	cloudbricks-sales-data	customerschanges/test/\$deadLetterQueue	subscription	peekLock	test_body	Succeeded

+ Add a custom footer

▼ Pages 4

▶ [Home](#)

▶ [Privacy statement](#)

▶ [Setup for your tenant](#)

▼ [User guide](#)

Setup and permissions

Setup for your tenant

Start the app and login

Provide Consent

Log out

Authenticate via user identity

Explore Service Bus Entities

Filter entities

Message counts

Select entities

Access Service Bus Messages

Browse Messages

Filtering

Show/Hide columns

Ordering

Apply limit

Show Message Details

Export Messages

Send Messages

Import Messages

messageId handling during import

Peek Messages

Receive Messages

Replay Messages

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/cloudbricksio/ServiceBusCloudExplorerSupport.wiki.git>

