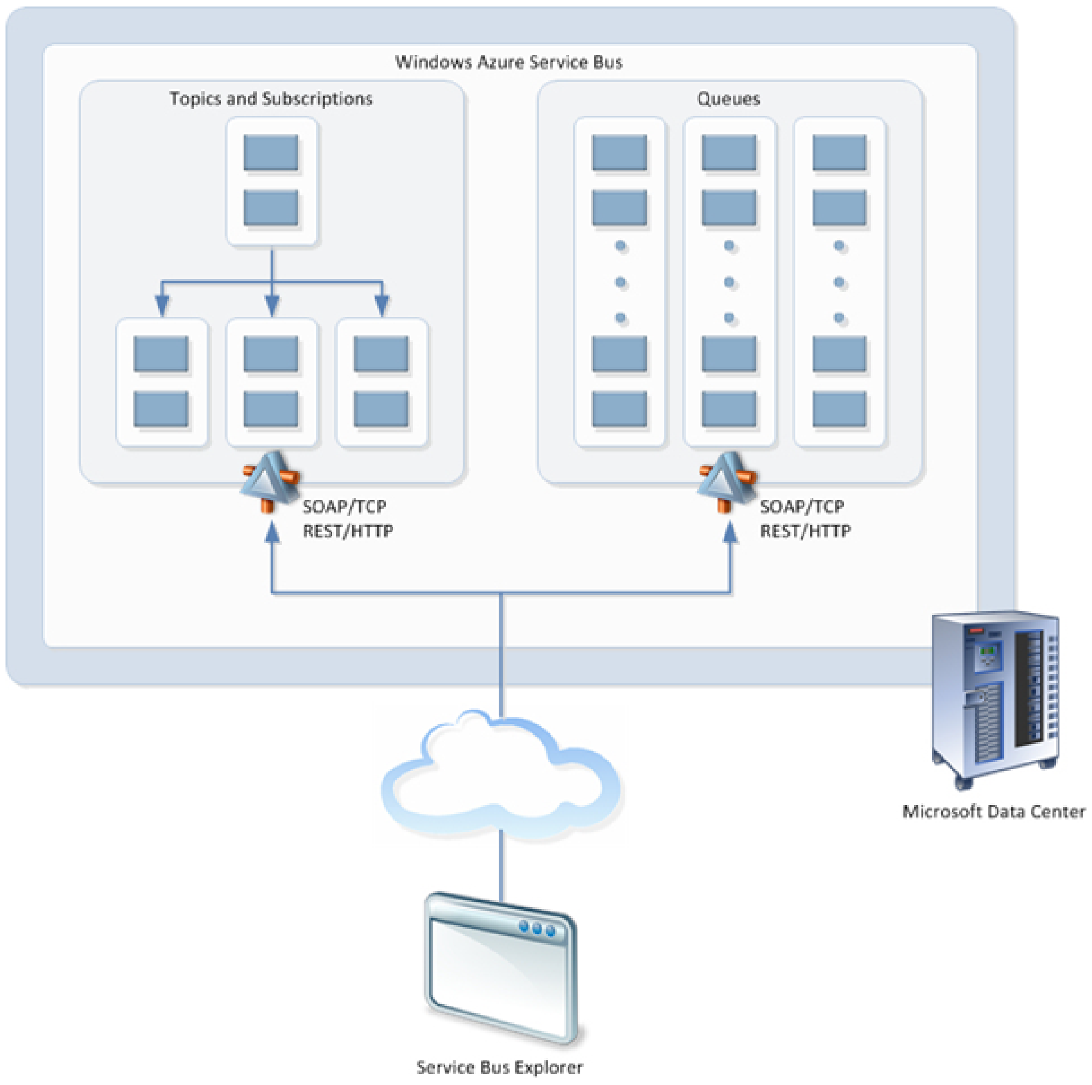
Service Bus Explorer

Created by TCS - Karne, Sudishna, last modified by TCS - Surya, Jai on Mar 30, 2020

The Service Bus Explorer allows users to connect to a **Service Bus namespace** and efficiently administer messaging entities. The tool provides advanced features like import/export functionality or the ability to test topic, queues, subscriptions, relay services, notification hubs and events hubs. With the help of Service Bus Explorer we can manage Service Bus resources.

The following picture illustrates the high-level architecture of the Service Bus Explorer tool.



In Azure, when we are working with Azure Service Bus where it has lots of functionalities but which are not available in Azure Portal. To overcome this there is tool called **Service Bus Explorer** which it has lot of functionalities to work with Azure Service Bus.

Azure Service Bus specific

These operations are specific to Azure Service Bus.

Connect to Azure Service Bus

Service Bus Explorer can connect to Azure Service Bus namespaces. In order to connect to Azure Service Bus go to Azure portal, open namespace we want, go to “Settings/Shared access policies”, and copy connection string from there. Finding out namespace tier (Basic/Standard/Premium) requires Azure Active Directory permissions, we have to provide service tier when creating a connection. This can be changed after connection is created.

Processing messages

In order to delete, schedule, defer, etc. message from a middle of a queue, Service Bus Explorer has to receive messages which come before wanted one, and to “Abandon” them afterwards. That increases Delivery Count of these messages, and if it reaches 10 (default, can be changed from queue properties), these messages will be deadlettered.

Service Bus Explorer will display warning dialog before operations which could affect DeliveryCount.

Deadlettering messages

We can move messages to a queue’s deadletter from context menu when one or more messages are selected.

We’ll be offered to enter **“Deadletter reason”** and **“Deadletter Error description”**. These are then added as a custom properties to deadlettered message, same as when Service Bus performs deadlettering internally.

Scheduled messages

We can schedule existing Azure Service Bus messages, i.e. delay their processing until specified time. Or we can reschedule already scheduled messages.

To schedule/reschedule message, select **“Schedule messages”** from context menu.

Deferred messages

We can defer processing of existing Azure Service Bus messages. To defer message(s), select **“Defer messages”** from context menu. After that operation messages will no longer be visible in their queue, but under “Deferred” subqueue. They will stay there until you manually do something with them.

Subscriptions

If you have to see all subscriptions from a topic, to delete or create multiple subscriptions, it’s best to click on a topic. It will display list of all its subqueues belonging to that topic.

Setting queue, topic, or subscription status

You can change status of Azure Service Bus queue, topic, or subscription to one of following:

- Active
- Disabled
- Receive Disabled
- Send Disabled

Comparison with Azure Service Bus Portal

What does Service Bus Explorer offer that is not available from Azure portal?

- **Preview of all queue messages** - see all messages from a queue or subscription.
- **Delete, Move or Copy** messages between queues or topics using drag&amp;drop or clipboard operations.
- **Save/load** - one or several messages can be selected at once and saved to a file. Messages can be loaded later to the same or another queue, on the same or different computer. This is great for backup purposes or for saving messages for later testing.
- **Access to deadletter queues** - check and fully manage deadletter queues.
- **Scheduled/deferred messages** - inspect and manage all scheduled or deferred messages in a queue.
- **Sessions** - access to all sessions in a queue, with messages inside them.
- **Subscription rules/filters** - inspect and adjust subscription rules and filters.
- **Sort messages** by any available field.
- **Filtering messages** by any field.
- **Connect to multiple namespaces**
- **Group namespaces**
- **Tabbed interface**

Message related features

- **Edit messages** - body and most important fields are editable.
- **Save/load of message bodies**
- **Create test messages** - easy creation of messages for testing purposes. Body can be loaded from file or entered in text field.
- **Additional message properties** - custom properties

Limitations when working with Azure Service Bus

Although Service Bus Explorer makes queues and messages look similar to folders and files, under the hood they are still first-in first-out queues. For some operations Service Bus Explorer will perform Receive and/or Send operations. Send operation will create new message and they’ll be put at the end of the queue.

General limitations

In order to delete, schedule, defer, etc. message from a middle of a queue, Service Bus Explorer has to receive messages which come before wanted one, and to “Abandon” them afterwards. That increases Delivery Count of these messages, and if it reaches 10 (default, can be changed from queue properties), these messages will be deadlettered.

When you load, copy, drag & drop or edit message it will go to the end of queue.

Since these messages are actually new messages in Azure Service Bus, some properties will be changed like time sent, time arrived, etc.

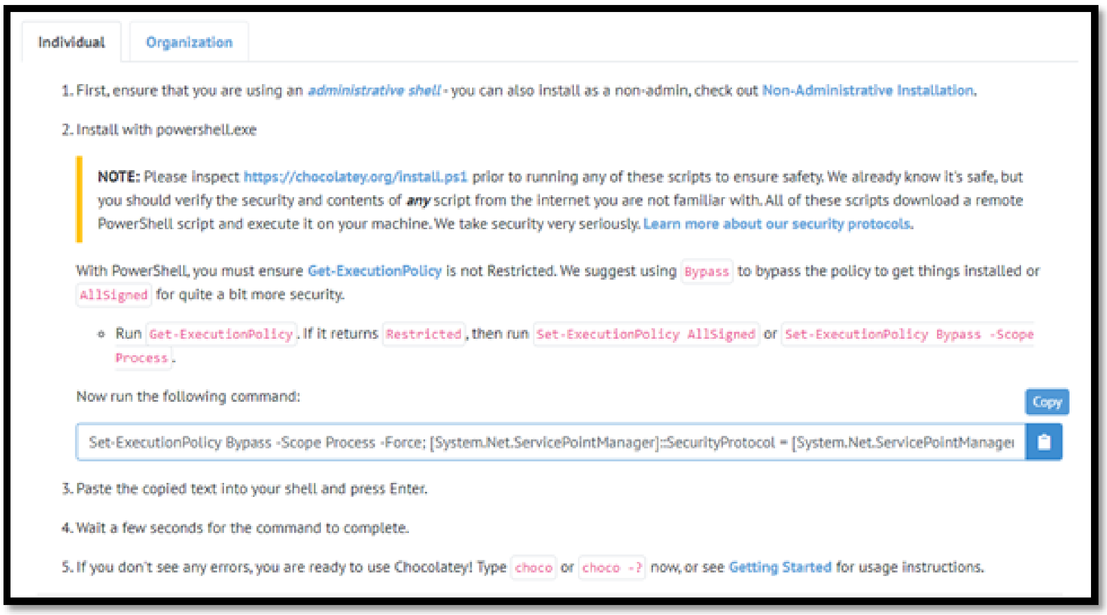
Service Bus Explorer doesn’t update display in real time. If some other application or another Service Bus Explorer window modifies some queue, you have to use “Refresh” to see these changes.

Some operations could fail or produce undesirable results if messages are removed in the mean time by some other program. It’s advisable to stop other processes that use queue while operations like copy, move, etc. are performed.

SERVICE BUS EXPLORER INSTALLATION

Using Chocolatey (a package manager for Windows) is the easiest way to install Service Bus Explorer.

Fig : 1.1



1. First, ensure that you are using an *administrative shell* - you can also install as a non-admin, check out Non-Administrative Installation.
2. Install with powershell.exe

   NOTE: Please inspect https://chocolatey.org/install.ps1 prior to running any of these scripts to ensure safety. We already know it's safe, but you should verify the security and contents of ***any*** script from the internet you are not familiar with. All of these scripts download a remote PowerShell script and execute it on your machine. We take security very seriously. Learn more about our security protocols.

   With PowerShell, you must ensure Get-ExecutionPolicy is not Restricted. We suggest using `Bypass` to bypass the policy to get things installed or `AllSigned` for quite a bit more security.

   - Run `Get-ExecutionPolicy`. If it returns `Restricted`, then run `Set-ExecutionPolicy AllSigned` or `Set-ExecutionPolicy Bypass -Scope Process`.

   Now run the following command:

   ```
   Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManage
   ```
3. Paste the copied text into your shell and press Enter.
4. Wait a few seconds for the command to complete.
5. If you don't see any errors, you are ready to use Chocolatey! Type `choco` or `choco -?` now, or see Getting Started for usage instructions.

Step 1: install Chocolatey

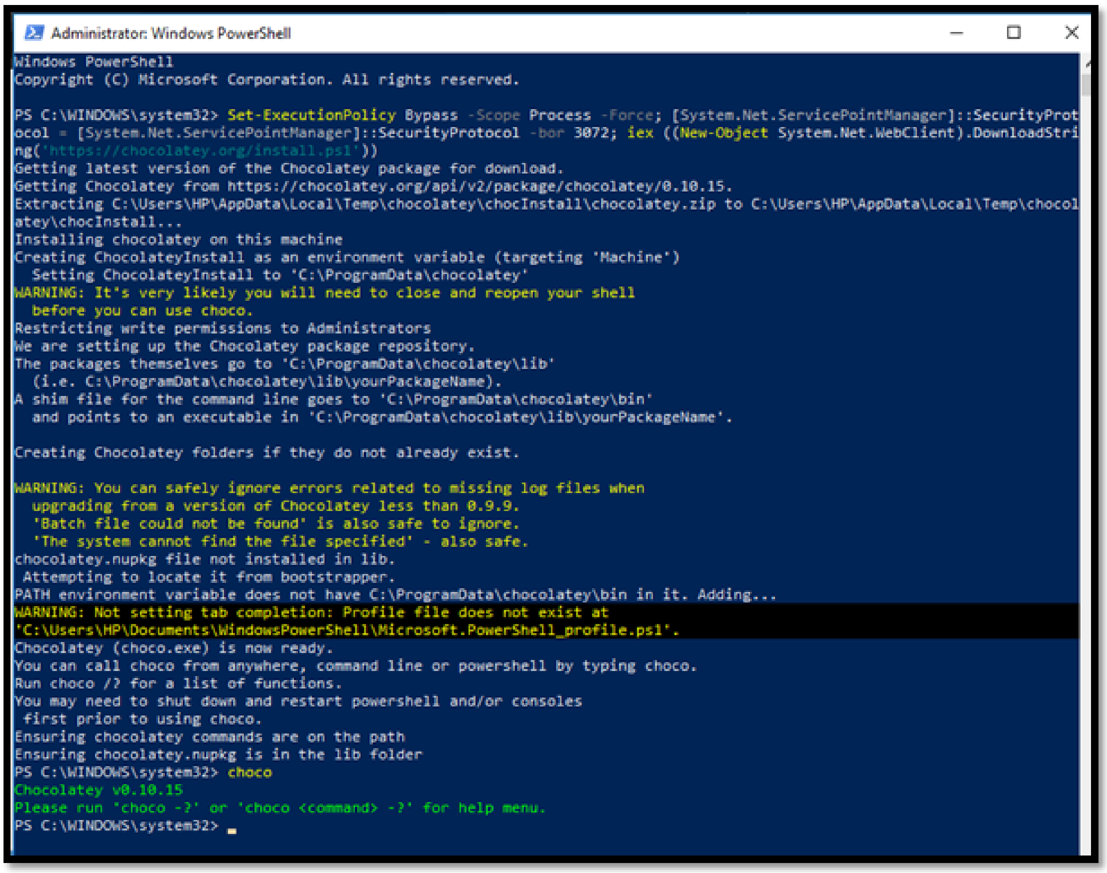
From the **Start** menu, run the **Windows PowerShell** application using the **Run as adminstrator** privileges.

Paste the PowerShell commands shown below in the into the PowerShell window and press **Enter**. Wait until the installation is finished.

Run the following command:

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

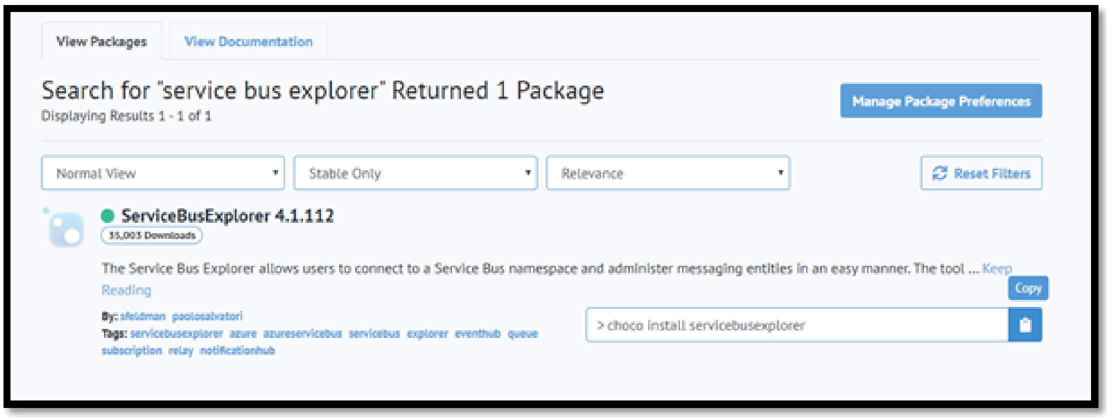
Fig : 1.2



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProt
ocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadStri
ng('https://chocolatey.org/install.ps1'))
Getting latest version of the Chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.15.
Extracting C:\Users\HP\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\HP\AppData\Local\Temp\chocol
atey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
  Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
  before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
  (i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
  upgrading from a version of Chocolatey less than 0.9.9.
  'Batch file could not be found' is also safe to ignore.
  'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
 Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\HP\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
 first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\WINDOWS\system32> choco
Chocolatey v0.10.15
Please run 'choco -?' or 'choco <command> -?' for help menu.
PS C:\WINDOWS\system32> _
```

Step 2: install Service Bus Explorer

Fig : 2.1



Search for "service bus explorer" Returned 1 Package
Displaying Results 1 - 1 of 1

ServiceBusExplorer 4.1.112
35,003 Downloads

The Service Bus Explorer allows users to connect to a Service Bus namespace and administer messaging entities in an easy manner. The tool ... Keep Reading

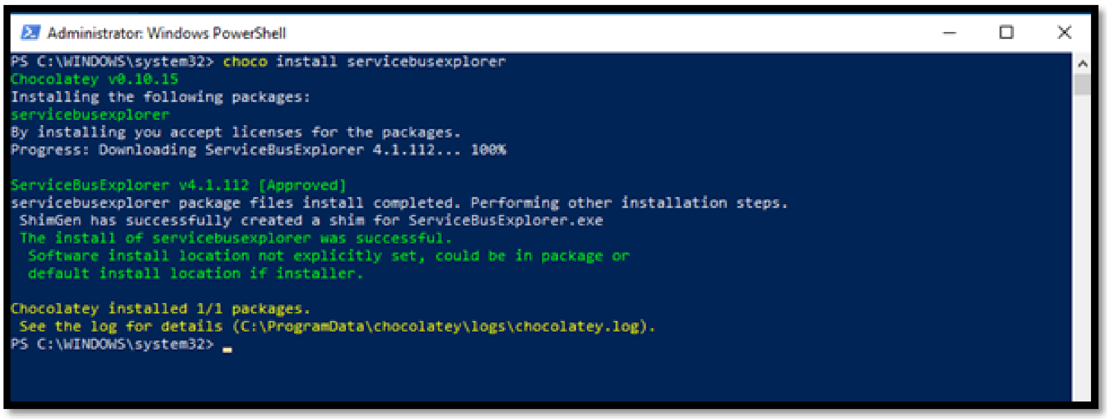
By: sfeldman paolosalvatori
Tags: servicebusexplorer azure azureservicebus servicebus explorer eventhub queue subscription relay notificationhub

```
> choco install servicebusexplorer
```

Run the following command:

choco install servicebusexplorer

fig : 2.2



```
PS C:\WINDOWS\system32> choco install servicebusexplorer
Chocolatey v0.10.15
Installing the following packages:
servicebusexplorer
By installing you accept licenses for the packages.
Progress: Downloading ServiceBusExplorer 4.1.112... 100%

ServiceBusExplorer v4.1.112 [Approved]
servicebusexplorer package files install completed. Performing other installation steps.
 ShimGen has successfully created a shim for ServiceBusExplorer.exe
 The install of servicebusexplorer was successful.
  Software install location not explicitly set, could be in package or
  default install location if installer.

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32> _
```

or

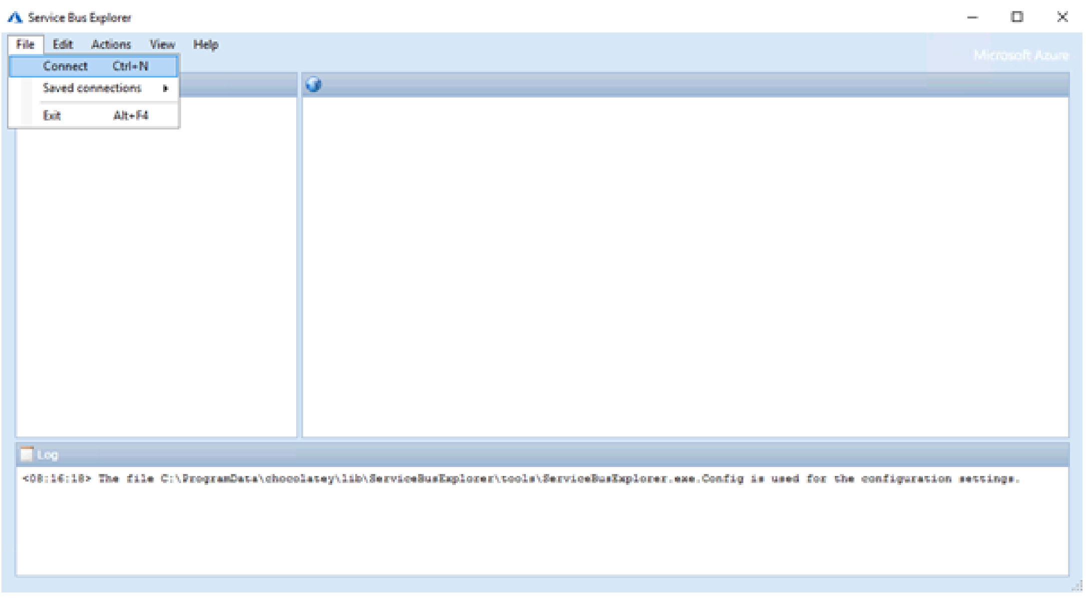
From the **Start** menu, start a **Command Prompt**. Paste the commands shown below in the into the Command Prompt window and press **Enter**. Wait until the installation is finished and close the Command Prompt window.

choco install servicebusexplorer

Connecting to a service bus namespace

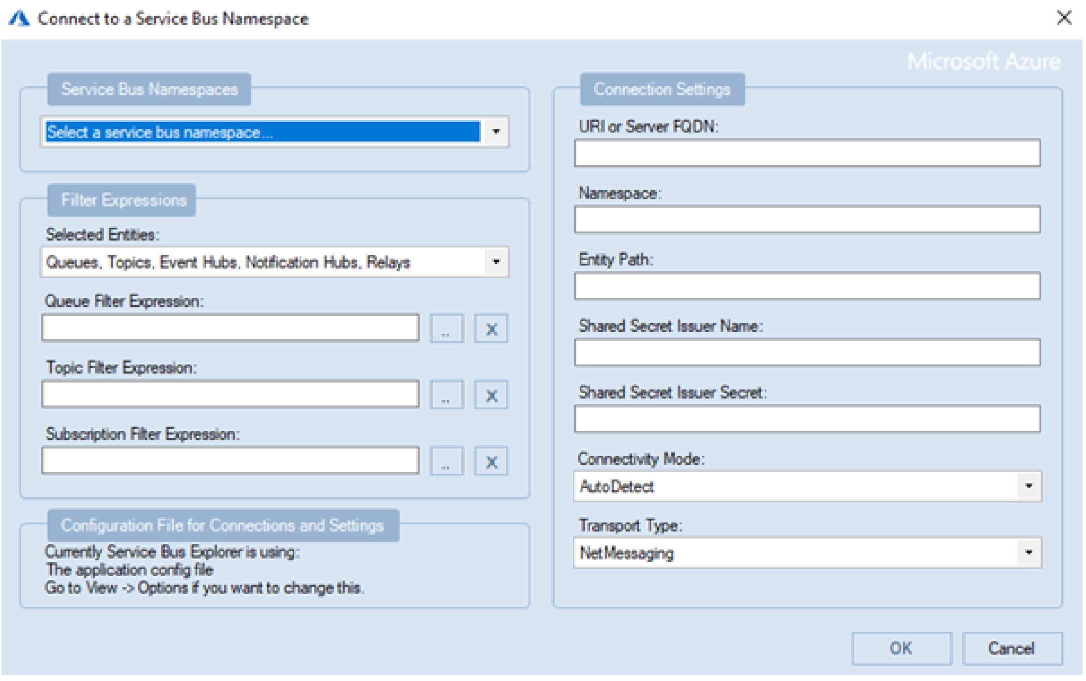
1. You can connect by entering all the required fields manually
2. You can connect by entering the full connection string
3. You can connect with the connection strings stored in the config file

Click on **File** >> **Connect**



```
<08:16:18> The file C:\ProgramData\chocolatey\lib\ServiceBusExplorer\tools\ServiceBusExplorer.exe.Config is used for the configuration settings.
```

Once you click on **Connect**, you will get a pop up window **Connect to Service Bus Namespace**



Select a service bus namespace...

Selected Entities: Queues, Topics, Event Hubs, Notification Hubs, Relays

Queue Filter Expression:

Topic Filter Expression:

Subscription Filter Expression:

Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.

URI or Server FQDN:

Namespace:

Entity Path:

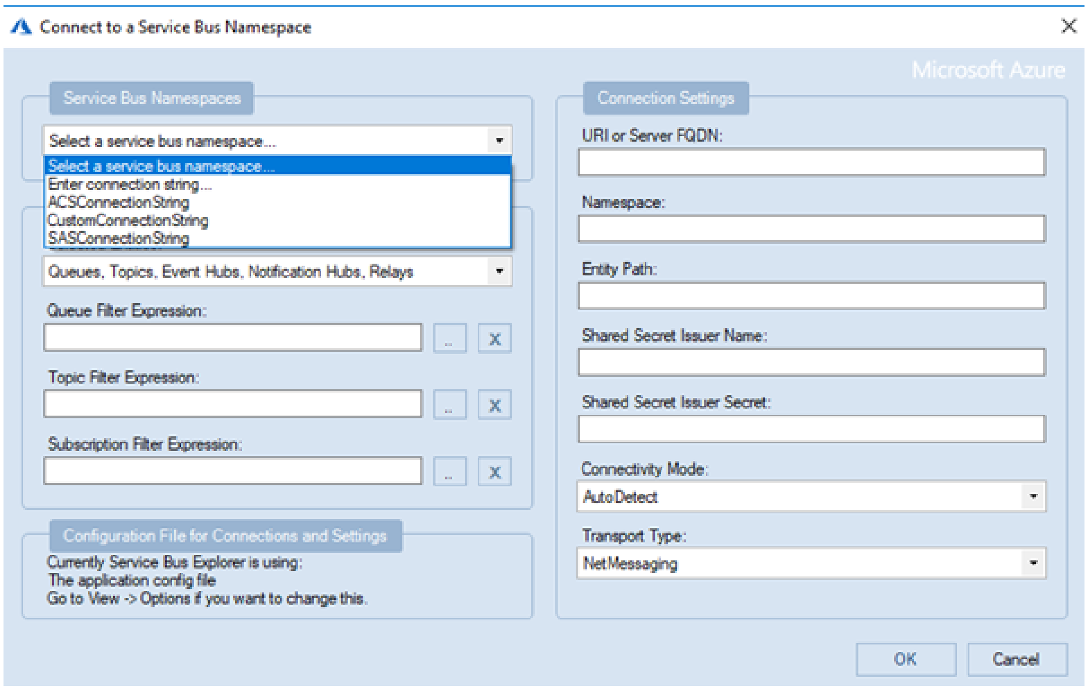
Shared Secret Issuer Name:

Shared Secret Issuer Secret:

Connectivity Mode: AutoDetect

Transport Type: NetMessaging

>> Under Service Bus Namespace, Select a **Service Bus namespace** drop down box


Connect to a Service Bus Namespace

Microsoft Azure

Service Bus Namespaces

Select a service bus namespace...
- Select a service bus namespace...
- Enter connection string...
- ACSConnectionString
- CustomConnectionString
- SASConnectionString

Queues, Topics, Event Hubs, Notification Hubs, Relays

Queue Filter Expression:

Topic Filter Expression:

Subscription Filter Expression:

Configuration File for Connections and Settings

Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.

Connection Settings

URI or Server FQDN:

Namespace:

Entity Path:

Shared Secret Issuer Name:

Shared Secret Issuer Secret:

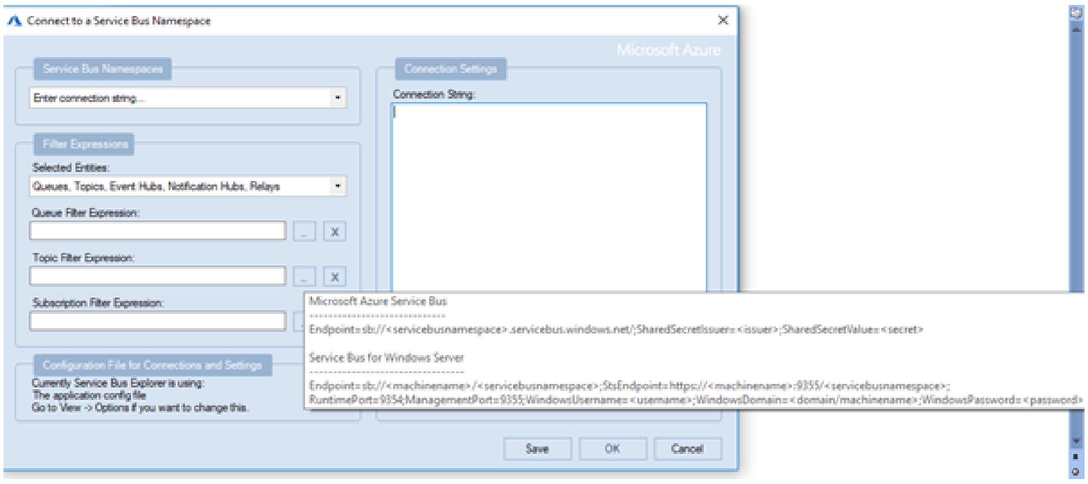
Connectivity Mode: AutoDetect

Transport Type: NetMessaging

>> get the Connection String from azure portal >>Service Bus >> Settings >> Shared Access Policies>> copy Connection String

>> Once you get the Connection String, Select **Enter connection string** from drop down box

>> paste the Connection String in **Connection String** dialog box


Connect to a Service Bus Namespace

Microsoft Azure

Service Bus Namespaces

Enter connection string...

Filter Expressions

Selected Entities: Queues, Topics, Event Hubs, Notification Hubs, Relays

Queue Filter Expression:

Topic Filter Expression:

Subscription Filter Expression:

Configuration File for Connections and Settings

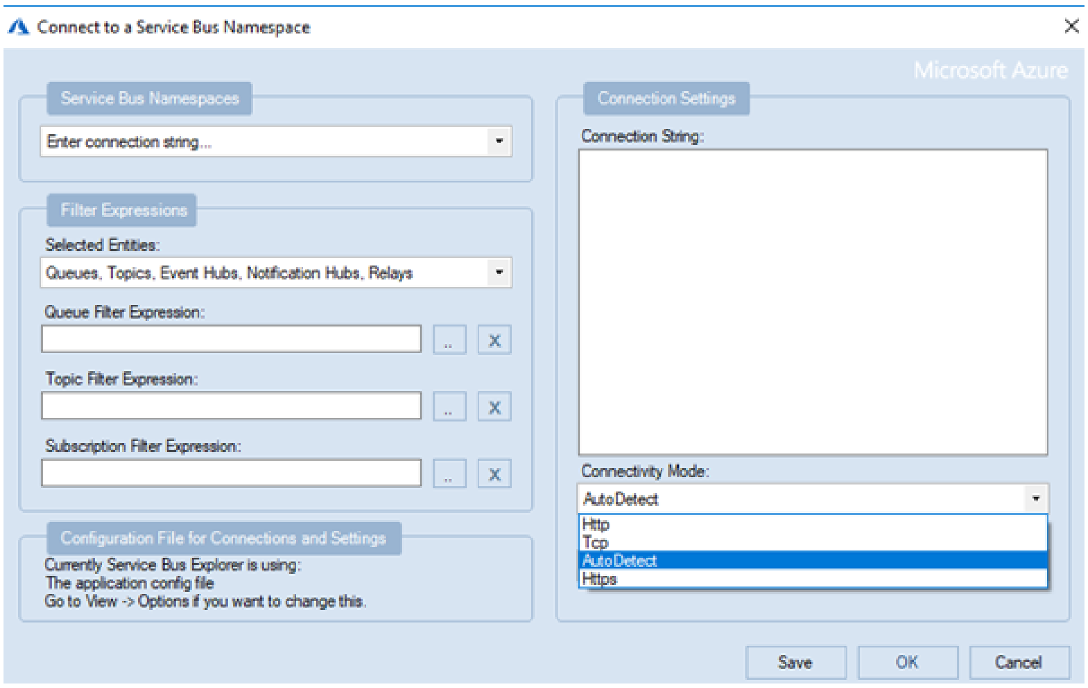
Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.

Connection Settings

Connection String:

```
Microsoft Azure Service Bus
------------------------------
Endpoint=sb://<servicebusnamespace>.servicebus.windows.net/;SharedSecretIssuer=<issuer>;SharedSecretValue=<secret>

Service Bus for Windows Server
------------------------------
Endpoint=sb://<machinename>/<servicebusnamespace>;StsEndpoint=https://<machinename>:9355/<servicebusnamespace>;
RuntimePort=9354;ManagementPort=9355;WindowsUsername=<username>;WindowsDomain=<domain/machinename>;WindowsPassword=<password>
```

>>Under Connectivity Mode>> Select Drop down box and select the **Connectivity Mode** you want


Connect to a Service Bus Namespace

Microsoft Azure

Service Bus Namespaces

Enter connection string...

Filter Expressions

Selected Entities: Queues, Topics, Event Hubs, Notification Hubs, Relays

Queue Filter Expression:

Topic Filter Expression:

Subscription Filter Expression:

Configuration File for Connections and Settings

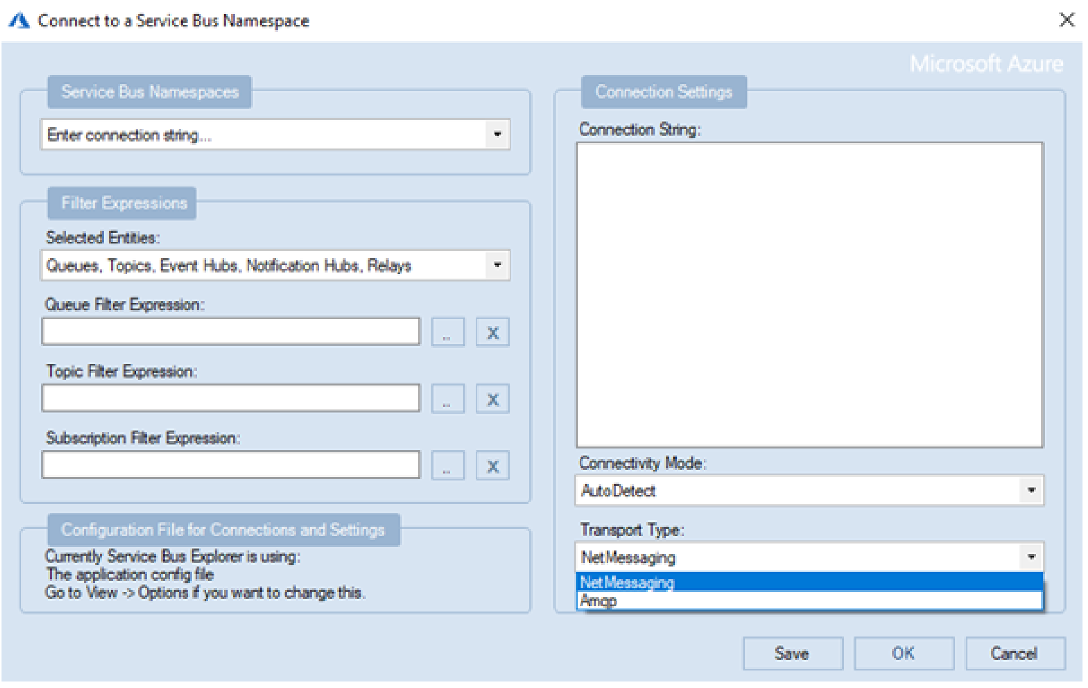
Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.

Connection Settings

Connection String:

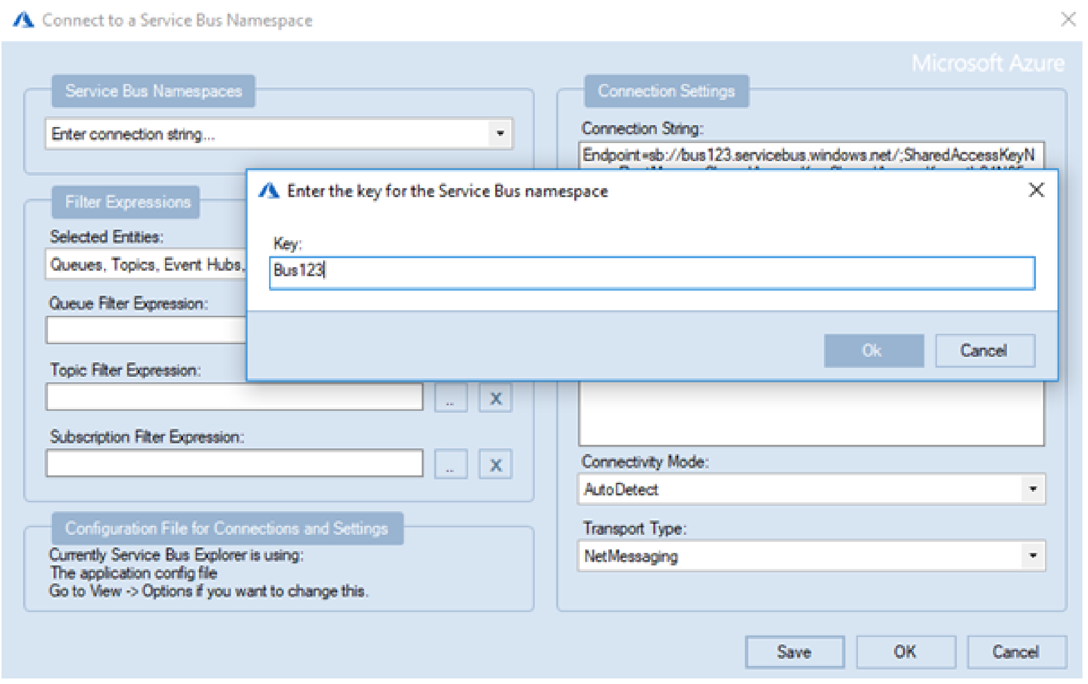
Connectivity Mode: AutoDetect
- Http
- Tcp
- AutoDetect
- Https

>> Under **Transport Type** >> Select Drop down box and select the **Transport Type** you want

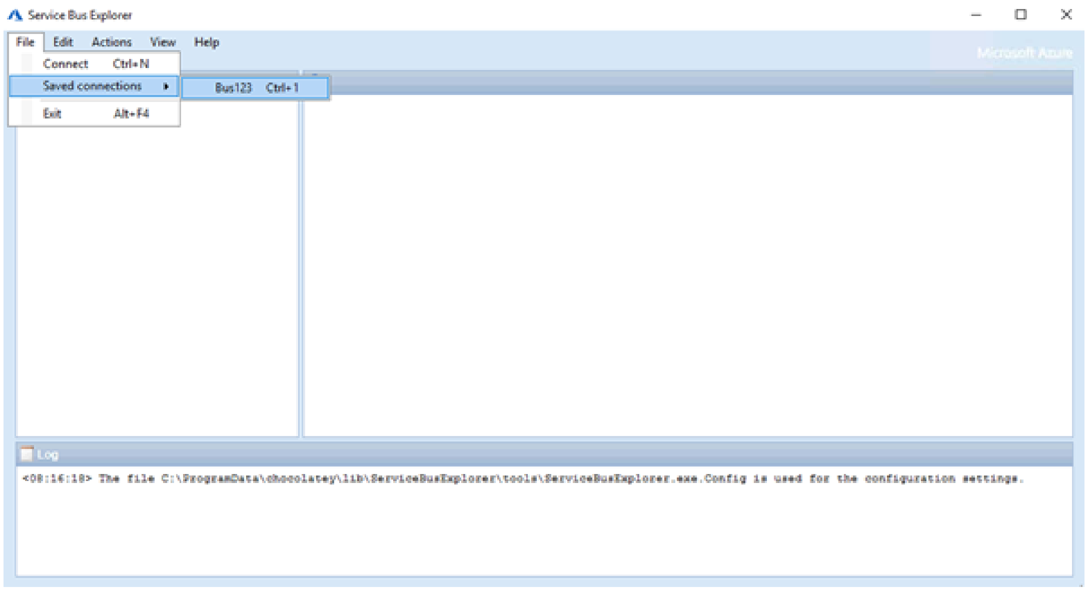

Connect to a Service Bus Namespace
Microsoft Azure
Service Bus Namespaces
Enter connection string...
Filter Expressions
Selected Entities:
Queues, Topics, Event Hubs, Notification Hubs, Relays
Queue Filter Expression:
Topic Filter Expression:
Subscription Filter Expression:
Configuration File for Connections and Settings
Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.
Connection Settings
Connection String:
Connectivity Mode:
AutoDetect
Transport Type:
NetMessaging
NetMessaging
Amqp
Save
OK
Cancel

>> click on **Save**

>> Enter the key for the Service Bus namespace >> click on **Ok**

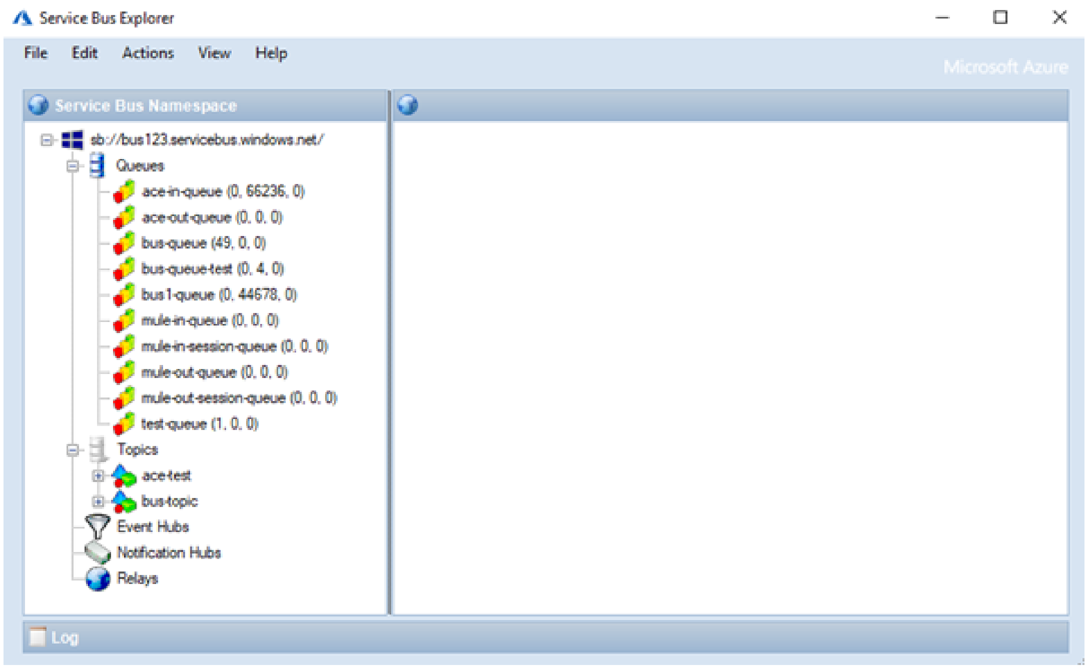

Connect to a Service Bus Namespace
Microsoft Azure
Service Bus Namespaces
Enter connection string...
Connection Settings
Connection String:
Endpoint=sb://bus123.servicebus.windows.net/;SharedAccessKeyN
Enter the key for the Service Bus namespace
Key:
Bus123
Ok
Cancel
Filter Expressions
Selected Entities:
Queues, Topics, Event Hubs,
Queue Filter Expression:
Topic Filter Expression:
Subscription Filter Expression:
Configuration File for Connections and Settings
Currently Service Bus Explorer is using:
The application config file
Go to View -> Options if you want to change this.
Connectivity Mode:
AutoDetect
Transport Type:
NetMessaging
Save
OK
Cancel

>> To open Saved Connections, Click on **File** >> **Saved Connections** >> Saved Service Bus Namespace (i.e., **Bus123**) to get started

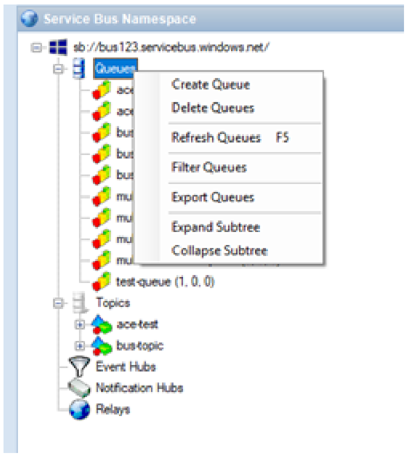


```
<08:16:18> The file C:\ProgramData\chocolatey\lib\ServiceBusExplorer\tools\ServiceBusExplorer.exe.Config is used for the configuration settings.
```

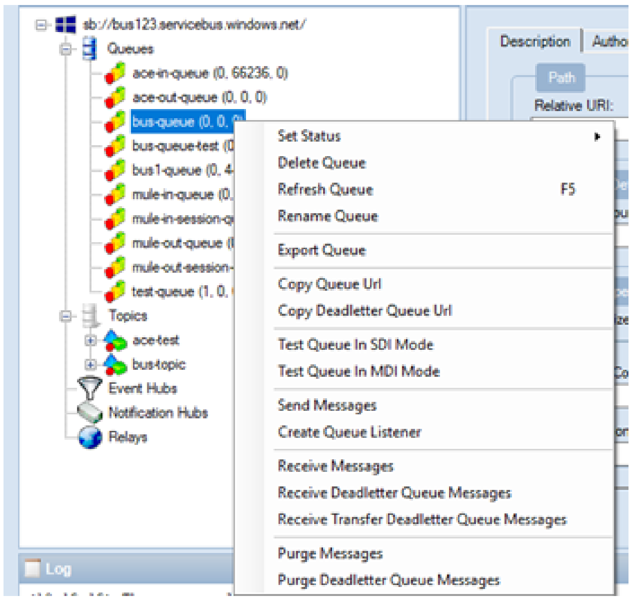
>> Once it is connected to service bus namespace, it appears all the resources of service bus queues and topics



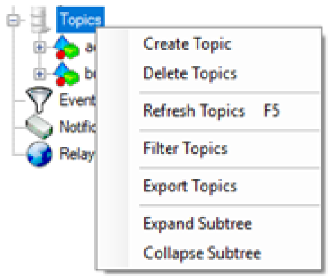
>> **Service Bus Namespace** >> When you Right Click on **Queues,** the following options will appear



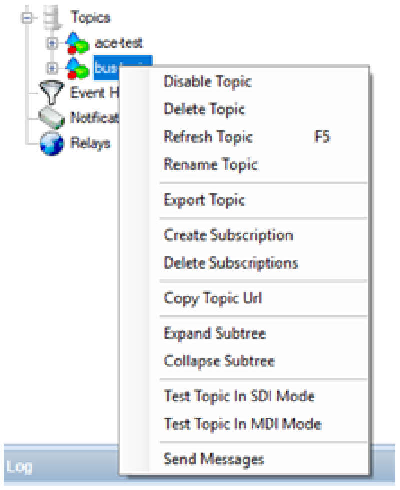
>> **Service Bus Namespace** >> **Queue** >>When you Right Click on **Queues name,** the following options will appear



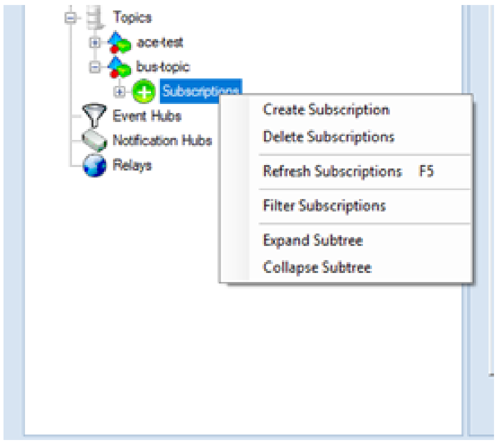
>> **Service Bus Namespace** >> When you Right Click on **Topic,** the following options will appear



>> **Service Bus Namespace** >> **Topic** >> When you Right Click on **Topic name,** the following options will appear

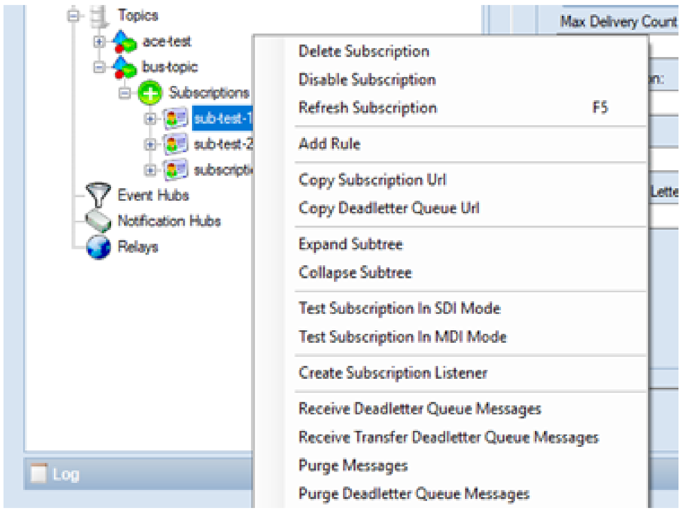


>> **Service Bus Namespace** >> **Topic** >> **Topic name** >> When you Right Click **Subscription**, the following options will appear



If you have to see all subscriptions from a topic, to delete or create multiple subscriptions, it’s best to click on a topic. It will display list of all its subqueues belonging to that topic.

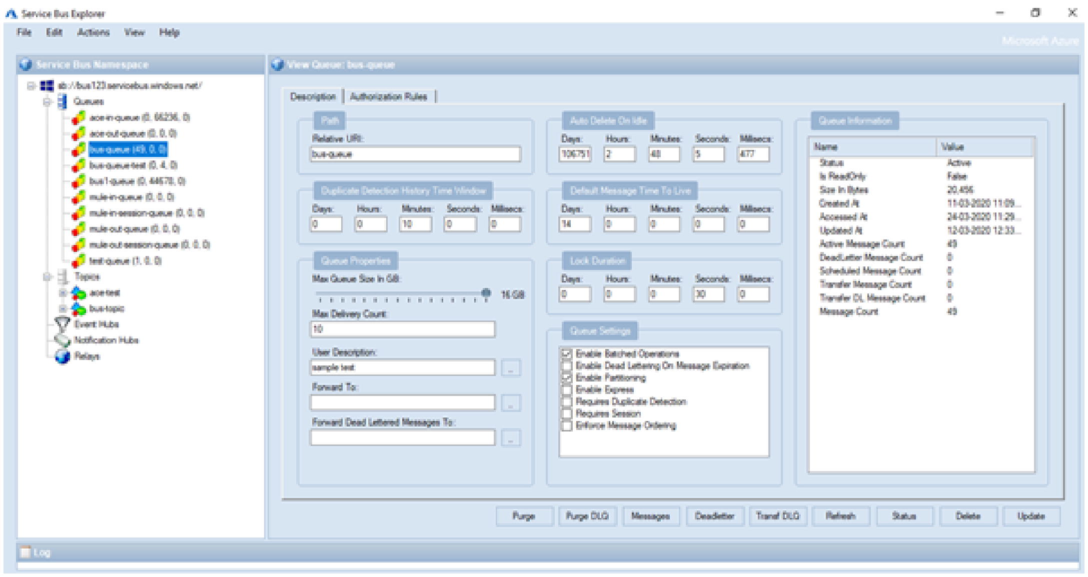
>> **Service Bus Namespace** >> **Topic** >> **Topic name** >> **Subscription** >> When you Right Click on **Subscriptions name**, the following options will appear



Message content and properties view

>> click on **Service Bus Namespace** resources to check the properties of corresponding queue/topic

>> In **View Queue** >> **Description**



Path
Relative URI: bus-queue

Auto Delete On Idle

Days:	Hours:	Minutes:	Seconds:	Millisecs:
106751	2	48	5	477

Duplicate Detection History Time Window

Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	10	0	0

Default Message Time To Live

Days:	Hours:	Minutes:	Seconds:	Millisecs:
14	0	0	0	0

Queue Properties
Max Queue Size In GB: 16 GB
Max Delivery Count: 10
User Description: sample text
Forward To:
Forward Dead Lettered Messages To:

Lock Duration

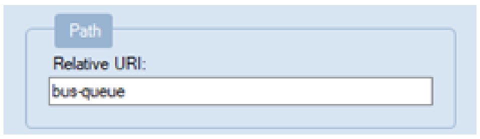
Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	0	30	0

Queue Settings
- ✓ Enable Batched Operations
- ☐ Enable Dead Lettering On Message Expiration
- ✓ Enable Partitioning
- ☐ Enable Express
- ☐ Requires Duplicate Detection
- ☐ Requires Session
- ☐ Enforce Message Ordering

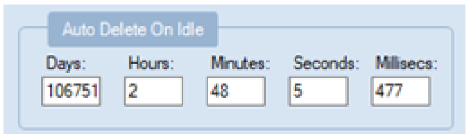
Queue Information

Name	Value
Status	Active
Is ReadOnly	False
Size In Bytes	20,456
Created At	11-03-2020 11:09...
Accessed At	24-03-2020 11:29...
Updated At	12-03-2020 12:33...
Active Message Count	49
DeadLetter Message Count	0
Scheduled Message Count	0
Transfer Message Count	0
Transfer DL Message Count	0
Message Count	49

You can set the **Relative URL** (i.e., Queue name or Topic name)

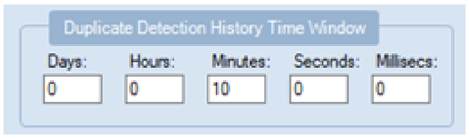

Path
Relative URI: bus-queue

You can set **Auto Delete On Idle**


Auto Delete On Idle

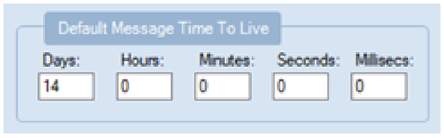
Days:	Hours:	Minutes:	Seconds:	Millisecs:
106751	2	48	5	477

You can set **Duplicate Deletion History Time Window**


Duplicate Detection History Time Window

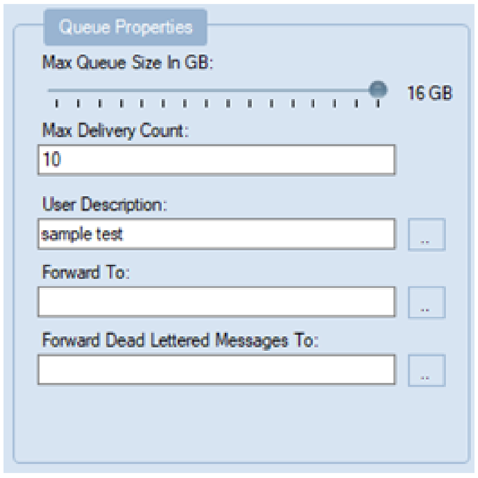
Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	10	0	0

You can set **Default Message Time To Live**



Days:	Hours:	Minutes:	Seconds:	Millisecs:
14	0	0	0	0

You can set **Queue Properties**



Max Queue Size In GB: 16 GB

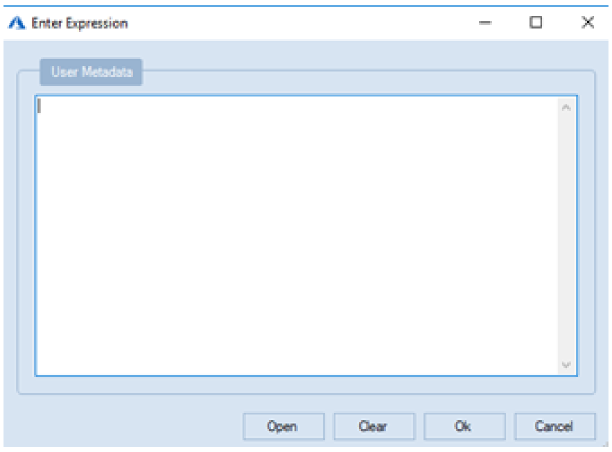
Max Delivery Count: 10

User Description: sample test

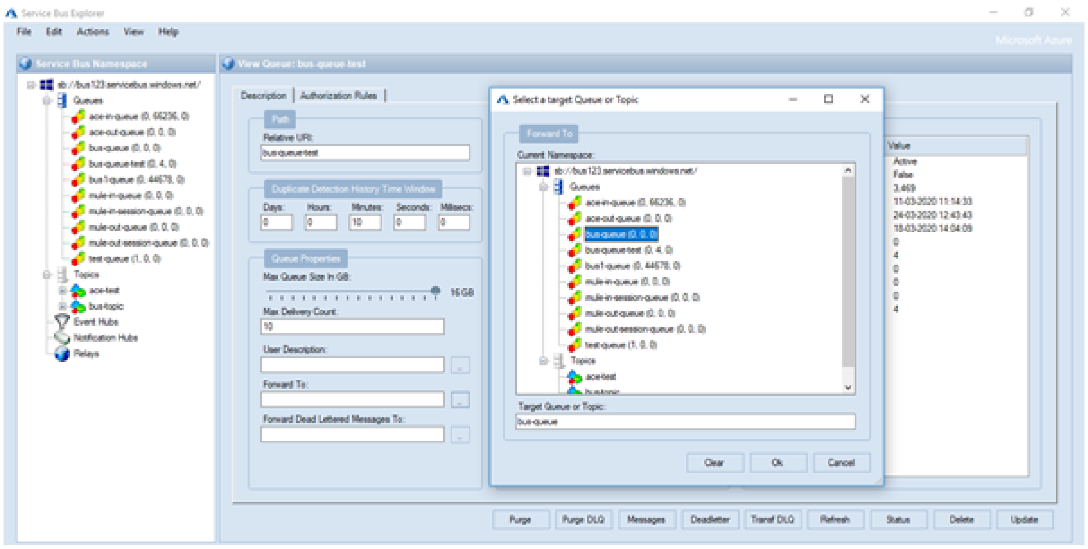
Forward To:

Forward Dead Lettered Messages To:

>>User Description



>> Forward To:



sb://bus123.servicebus.windows.net/
- Queues
  - ace-in-queue (0, 66236, 0)
  - ace-out-queue (0, 0, 0)
  - bus-queue (0, 0, 0)
  - bus-queue-test (0, 4, 0)
  - bus1-queue (0, 44678, 0)
  - mule-in-queue (0, 0, 0)
  - mule-in-session-queue (0, 0, 0)
  - mule-out-queue (0, 0, 0)
  - mule-out-session-queue (0, 0, 0)
  - test-queue (1, 0, 0)
- Topics
  - ace-test
  - bus-topic
- Event Hubs
- Notification Hubs
- Relays

Path

Relative URI: bus-queue-test

Duplicate Detection History Time Window

Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	10	0	0

Queue Properties

Max Queue Size In GB: 16 GB

Max Delivery Count: 10

User Description:

Forward To:

Forward Dead Lettered Messages To:

Select a target Queue or Topic:

Forward To

Current Namespace:

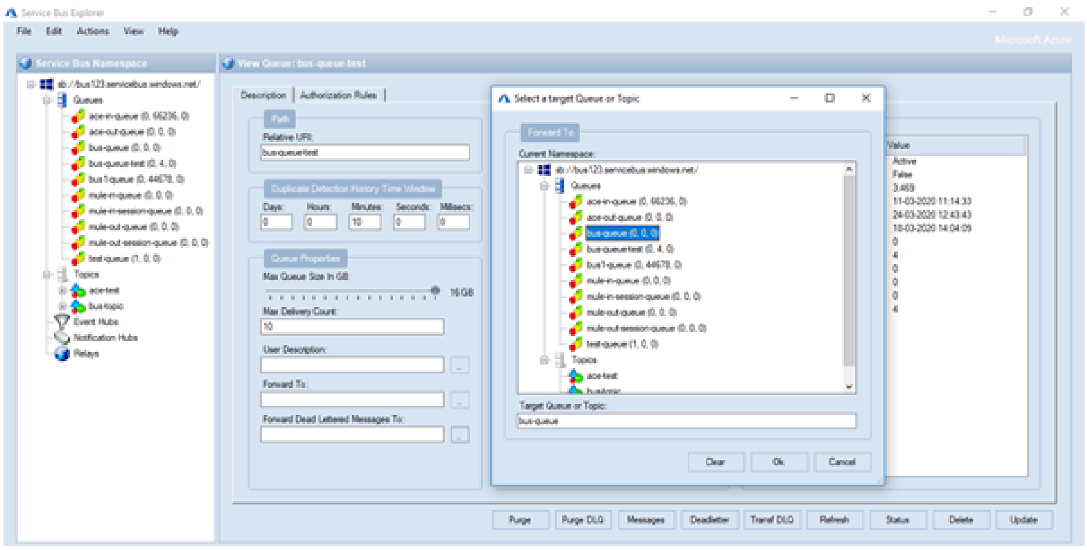
sb://bus123.servicebus.windows.net/
- Queues
  - ace-in-queue (0, 66236, 0)
  - ace-out-queue (0, 0, 0)
  - bus-queue (0, 0, 0)
  - bus-queue-test (0, 4, 0)
  - bus1-queue (0, 44678, 0)
  - mule-in-queue (0, 0, 0)
  - mule-in-session-queue (0, 0, 0)
  - mule-out-queue (0, 0, 0)
  - mule-out-session-queue (0, 0, 0)
  - test-queue (1, 0, 0)
- Topics
  - ace-test
  - bus-topic

Target Queue or Topic: bus-queue

Clear Ok Cancel

Value
Active
False
3,469
11-03-2020 11:14:33
24-03-2020 12:43:43
18-03-2020 14:04:09
0
4
0
0
0
4

>>Forward Dead Lettered Messages To:



sb://bus123.servicebus.windows.net/
Queues
ace-in-queue (0, 66236, 0)
ace-out-queue (0, 0, 0)
bus-queue (0, 0, 0)
bus-queue-test (0, 4, 0)
bus1queue (0, 44678, 0)
mule-in-queue (0, 0, 0)
mule-in-session-queue (0, 0, 0)
mule-out-queue (0, 0, 0)
mule-out-session-queue (0, 0, 0)
test-queue (1, 0, 0)
Topics
ace-test
bus-topic
Event Hubs
Notification Hubs
Relays

Path
Relative URI: bus-queue-test

Duplicate Detection History Time Window

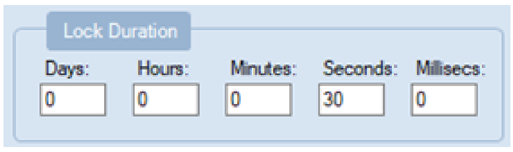
Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	10	0	0

Queue Properties
Max Queue Size In GB: 16 GB
Max Delivery Count: 10
User Description:
Forward To:
Forward Dead Lettered Messages To:

Select a target Queue or Topic
Forward To
Current Namespace:
sb://bus123.servicebus.windows.net/
Queues
ace-in-queue (0, 66236, 0)
ace-out-queue (0, 0, 0)
bus-queue (0, 0, 0)
bus-queue-test (0, 4, 0)
bus1queue (0, 44678, 0)
mule-in-queue (0, 0, 0)
mule-in-session-queue (0, 0, 0)
mule-out-queue (0, 0, 0)
mule-out-session-queue (0, 0, 0)
test-queue (1, 0, 0)
Topics
ace-test
bus-topic
Target Queue or Topic: bus-queue

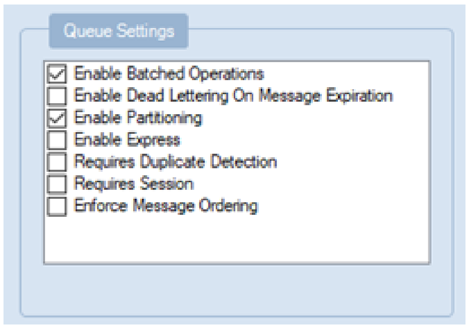
Value
Active
False
3,469
11-03-2020 11:14:33
24-03-2020 12:43:43
18-03-2020 14:04:09
0
4
0
0
0
4

You can set **Lock Duration**


Lock Duration

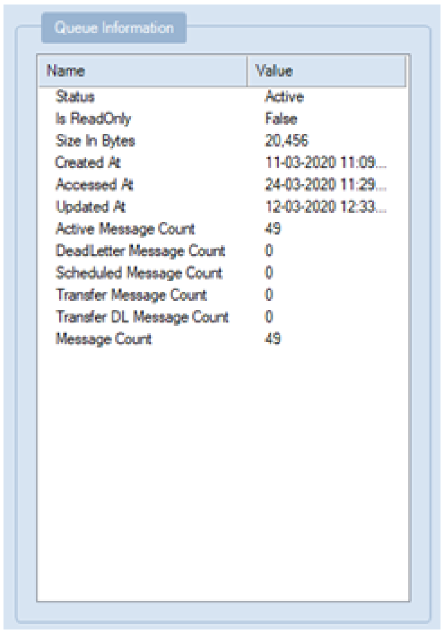
Days:	Hours:	Minutes:	Seconds:	Millisecs:
0	0	0	30	0

You can set **Queue Settings**


Queue Settings

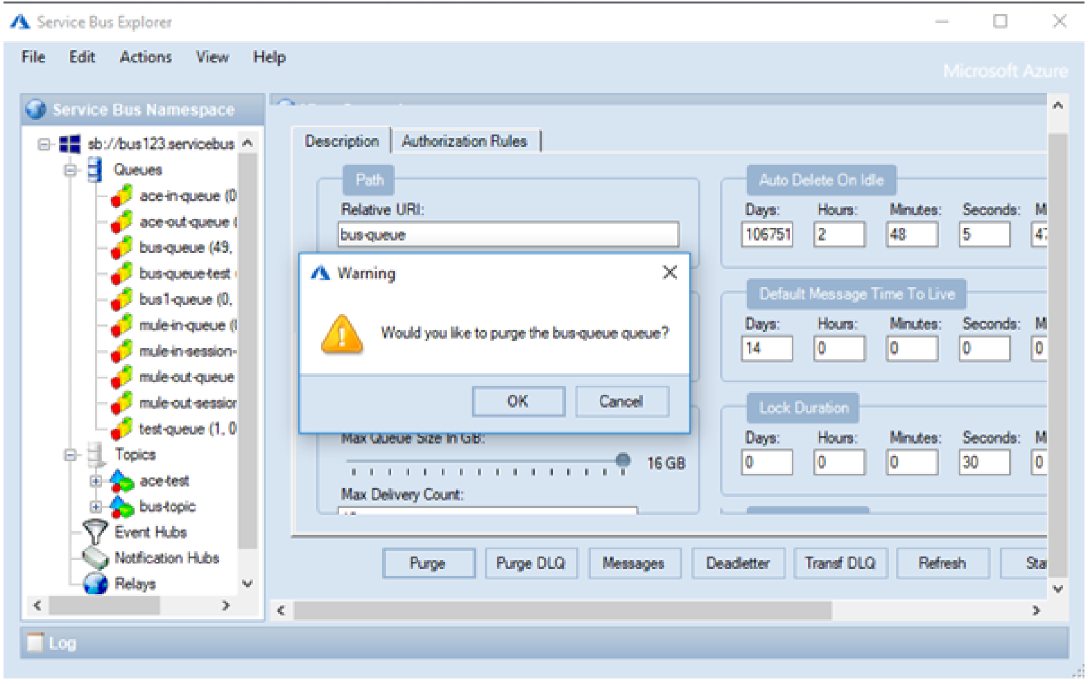
- [x] Enable Batched Operations
- [ ] Enable Dead Lettering On Message Expiration
- [x] Enable Partitioning
- [ ] Enable Express
- [ ] Requires Duplicate Detection
- [ ] Requires Session
- [ ] Enforce Message Ordering

>>View Queue Information


Queue Information

Name	Value
Status	Active
Is ReadOnly	False
Size In Bytes	20,456
Created At	11-03-2020 11:09...
Accessed At	24-03-2020 11:29...
Updated At	12-03-2020 12:33...
Active Message Count	49
DeadLetter Message Count	0
Scheduled Message Count	0
Transfer Message Count	0
Transfer DL Message Count	0
Message Count	49

>> When you Click on **Purge**, a **warning** pop up window will appear

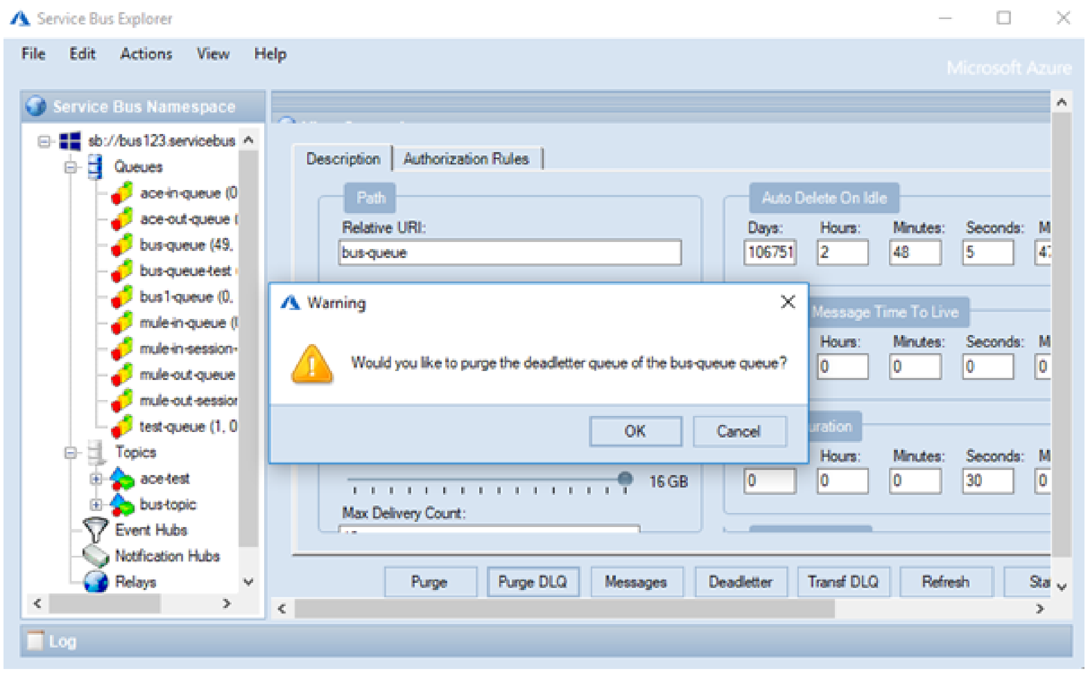


Warning

Would you like to purge the bus-queue queue?

OK Cancel

>> When you Click on **Purge DLQ**(DeadLetterQueue), a **warning** pop up window will appear



Warning

Would you like to purge the deadletter queue of the bus-queue queue?

OK Cancel

>> When you Click on **Messages**

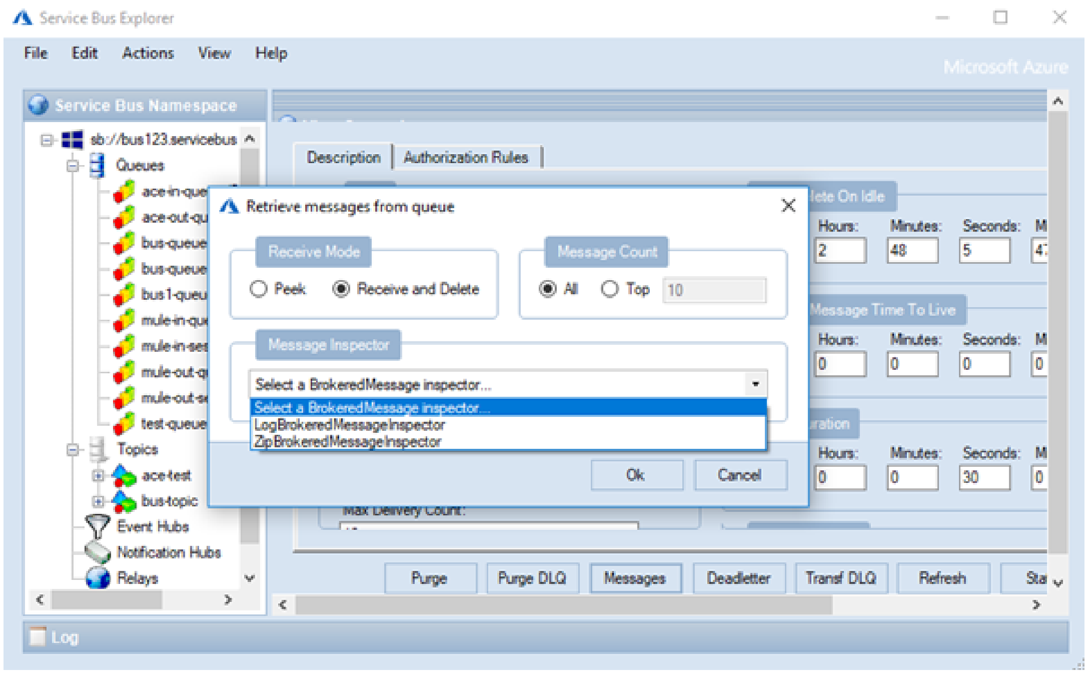
>> **Retrieve messages from queue** pop up Window will appear

>> Under **Receive Mode** >> Select radio button either **Peek** or **Receive and Delete**

>> Under **Message Count >>** Select radio button either **All** or **Top** (Count of messages in dialog box)

>> Under **Message Inspector >>** Select a **BrokeredMessage inspector** from drop down box (Optional)

>> Click on **Ok**


Service Bus Explorer
File Edit Actions View Help
Service Bus Namespace
Description Authorization Rules
Purge Purge DLQ Messages Deadletter Transf DLQ Refresh
Log

Retrieve messages from queue

Receive Mode: ( ) Peek [✓ (•) Receive and Delete]

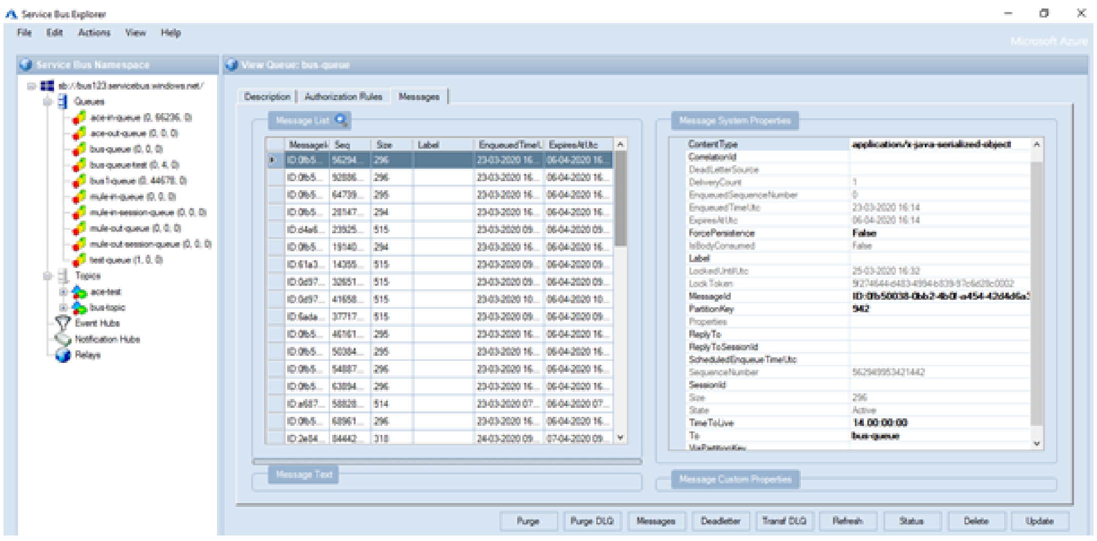
Message Count: [✓ (•) All] ( ) Top 10

Message Inspector: Select a BrokeredMessage inspector...
- Select a BrokeredMessage inspector...
- LogBrokeredMessageInspector
- ZipBrokeredMessageInspector

Ok Cancel

>> After Clicking on **Ok ,** the following will get appeared

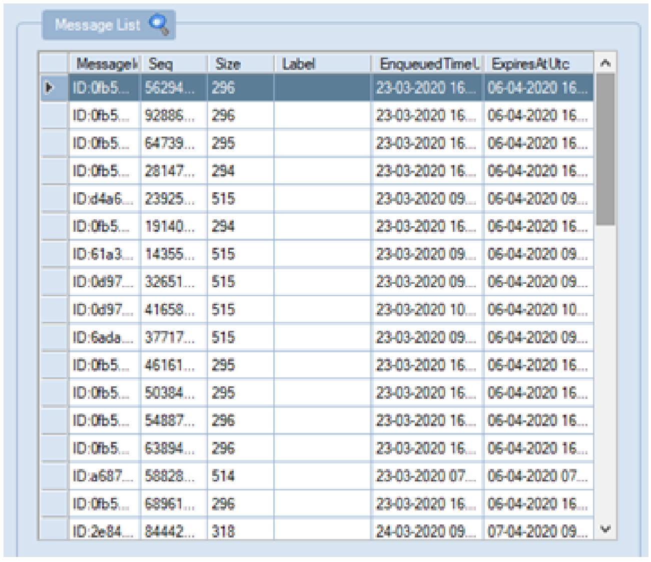
>> Message List


Service Bus Explorer
File Edit Actions View Help
Service Bus Namespace
View Queue: bus-queue
Description Authorization Rules Messages
Message List
Message Text
Message Custom Properties
Purge Purge DLQ Messages Deadletter Transf DLQ Refresh Status Delete Update

Message System Properties

ContentType	application/x-java-serialized-object
CorrelationId	
DeadLetterSource	
DeliveryCount	1
EnqueuedSequenceNumber	0
EnqueuedTimeUtc	23-03-2020 16:14
ExpiresAtUtc	06-04-2020 16:14
ForcePersistence	False
IsBodyConsumed	False
Label	
LockedUntilUtc	25-03-2020 16:32
Lock Token	9f274644-d483-4994-b839-97e6d28c0002
MessageId	ID:0fb50038-0bb2-4b0f-a454-42d4d6a3
PartitionKey	942
Properties	
ReplyTo	
ReplyToSessionId	
ScheduledEnqueueTimeUtc	
SequenceNumber	562949953421442
SessionId	
Size	296
State	Active
TimeToLive	14.00:00:00
To	bus-queue

Here, in the below figure you can see all the messages **with Message Id, Sequence number, Size, Label, EnqueuedTime, ExpiresAt.**

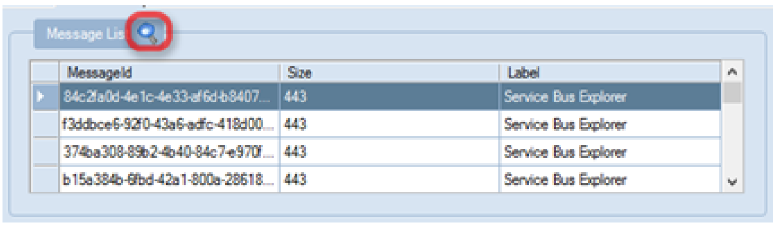


Message List

MessageI	Seq	Size	Label	EnqueuedTimeL	ExpiresAtUtc
ID:0fb5...	56294...	296		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	92886...	296		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	64739...	295		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	28147...	294		23-03-2020 16...	06-04-2020 16...
ID:d4a6...	23925...	515		23-03-2020 09...	06-04-2020 09...
ID:0fb5...	19140...	294		23-03-2020 16...	06-04-2020 16...
ID:61a3...	14355...	515		23-03-2020 09...	06-04-2020 09...
ID:0d97...	32651...	515		23-03-2020 09...	06-04-2020 09...
ID:0d97...	41658...	515		23-03-2020 10...	06-04-2020 10...
ID:6ada...	37717...	515		23-03-2020 09...	06-04-2020 09...
ID:0fb5...	46161...	295		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	50384...	295		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	54887...	296		23-03-2020 16...	06-04-2020 16...
ID:0fb5...	63894...	296		23-03-2020 16...	06-04-2020 16...
ID:a687...	58828...	514		23-03-2020 07...	06-04-2020 07...
ID:0fb5...	68961...	296		23-03-2020 16...	06-04-2020 16...
ID:2e84...	84442...	318		24-03-2020 09...	07-04-2020 09...

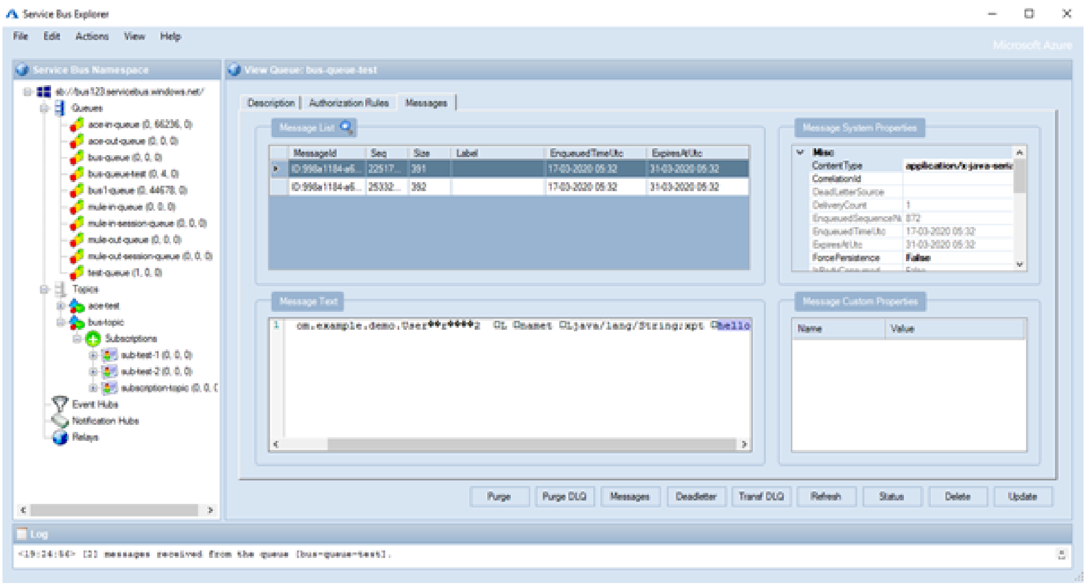
Filter Expression

Click the magnifying glass button to define a filter expression for received messages using a SQL Expression (e.g. sys.Size > 300 and sys.Label='Service Bus Explorer').



MessageId	Size	Label
84c2fa0d-4e1c-4e33-af6d-b8407...	443	Service Bus Explorer
f3ddbce6-92f0-43a6-adfc-418d00...	443	Service Bus Explorer
374ba308-89b2-4b40-84c7-e970f...	443	Service Bus Explorer
b15a384b-6fbd-42a1-800a-28618...	443	Service Bus Explorer

Repair and Resubmit Message



MessageId	Seq	Size	Label	EnqueuedTimeUtc	ExpiresAtUtc
ID:998a1184-a6...	22517...	391		17-03-2020 05:32	31-03-2020 05:32
ID:998a1184-a6...	25332...	392		17-03-2020 05:32	31-03-2020 05:32

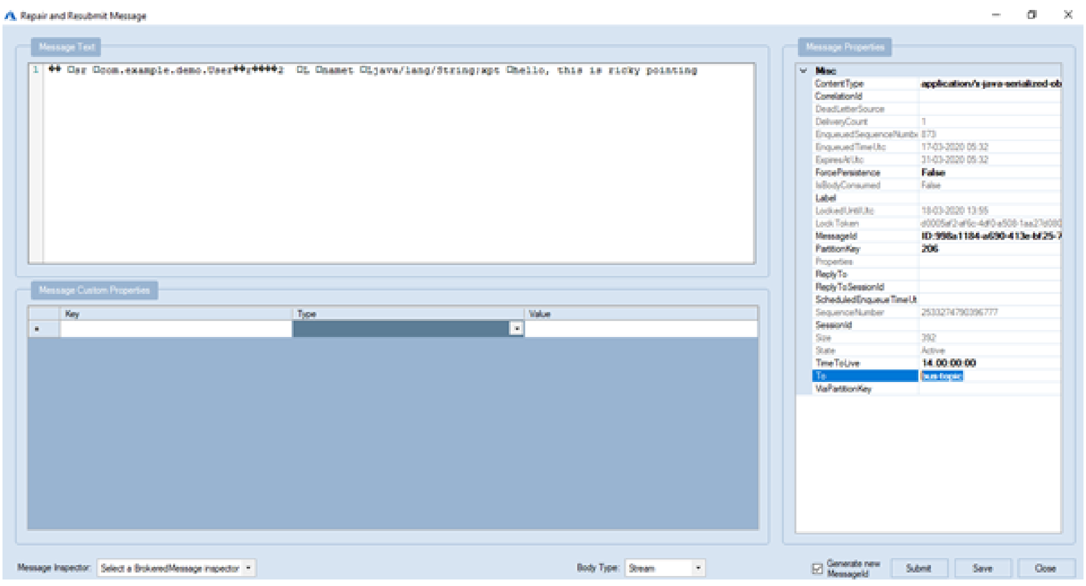
Misc	
ContentType	application/x-java-seria
CorrelationId	
DeadLetterSource	
DeliveryCount	1
EnqueuedSequenceN	872
EnqueuedTimeUtc	17-03-2020 05:32
ExpiresAtUtc	31-03-2020 05:32
ForcePersistence	False

```
om.example.demo.User��r����2  �L �namet �Ljava/lang/String;xpt �hello
```

Name	Value

```
<19:24:56> [2] messages received from the queue [bus-queue-test].
```

Double click a row or click **Repair and Resubmit Message** from the context menu to open a message in a separate dialog. This functionality allows to clone and send the selected messages to the same or an alternative queue or topic in the **Service Bus** namespace. If you want to edit the payload, system properties or user-defined properties, you have to select, edit and send messages one at a time.

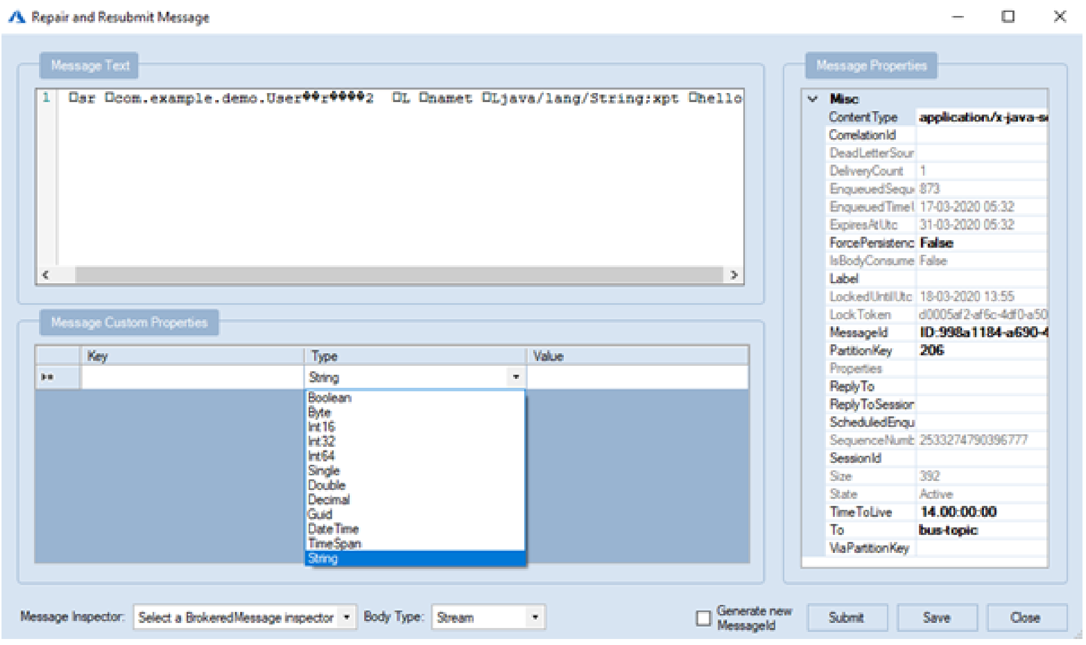


```
�� �sr �com.example.demo.User��r����2  �L �namet �Ljava/lang/String;xpt �hello, this is ricky pointing
```

Key	Type	Value

Misc	
ContentType	application/x-java-serialized-ob
CorrelationId	
DeadLetterSource	
DeliveryCount	1
EnqueuedSequenceNumbe	873
EnqueuedTimeUtc	17-03-2020 05:32
ExpiresAtUtc	31-03-2020 05:32
ForcePersistence	False
IsBodyConsumed	False
Label	
LockedUntilUtc	18-03-2020 13:55
Lock Token	d0005af2-af6c-4df0-a508-1aa27d080
MessageId	ID:998a1184-a690-413e-bf25-7
PartitionKey	206
Properties	
ReplyTo	
ReplyToSessionId	
ScheduledEnqueueTimeUt	
SequenceNumber	2533274790396777
SessionId	
Size	392
State	Active
TimeToLive	14.00:00:00
To	bus-topic
ViaPartitionKey	

>> Custom Message Properties



```
Dsr Dcom.example.demo.User��r����2  DL Dnamet DLjava/lang/String;xpt Dhello
```

	Key	Type	Value
▸*		String	

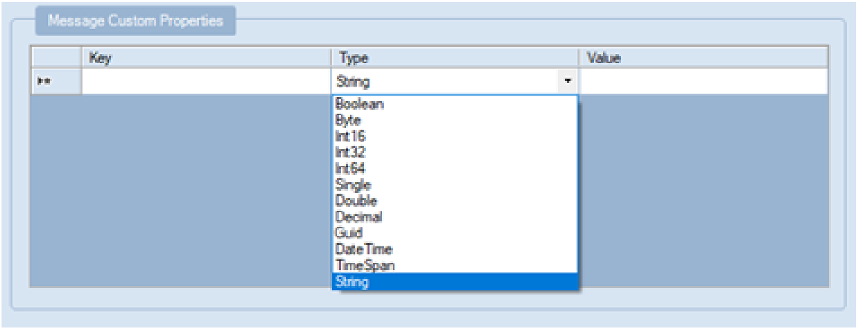
Boolean, Byte, Int16, Int32, Int64, Single, Double, Decimal, Guid, DateTime, TimeSpan, String

Misc	
ContentType	application/x-java-se
CorrelationId	
DeadLetterSour	
DeliveryCount	1
EnqueuedSequ	873
EnqueuedTimeU	17-03-2020 05:32
ExpiresAtUtc	31-03-2020 05:32
ForcePersistenc	False
IsBodyConsume	False
Label	
LockedUntilUtc	18-03-2020 13:55
LockToken	d0005af2-af6c-4df0-a50
MessageId	ID:998a1184-a690-4
PartitionKey	206
Properties	
ReplyTo	
ReplyToSession	
ScheduledEnqu	
SequenceNumb	25332747903967777
SessionId	
Size	392
State	Active
TimeToLive	14.00:00:00
To	bus-topic
ViaPartitionKey	

Message Inspector: Select a BrokeredMessage inspector   Body Type: Stream   ☐ Generate new MessageId

Message related features

- **Edit messages** - body and most important fields are editable.
- **Save/load of message bodies**
- **Create test messages** - easy creation of messages for testing purposes. Body can be loaded from file or entered in text field
- **Additional message properties** - custom properties



	Key	Type	Value
▸*		String	

Boolean, Byte, Int16, Int32, Int64, Single, Double, Decimal, Guid, DateTime, TimeSpan, String

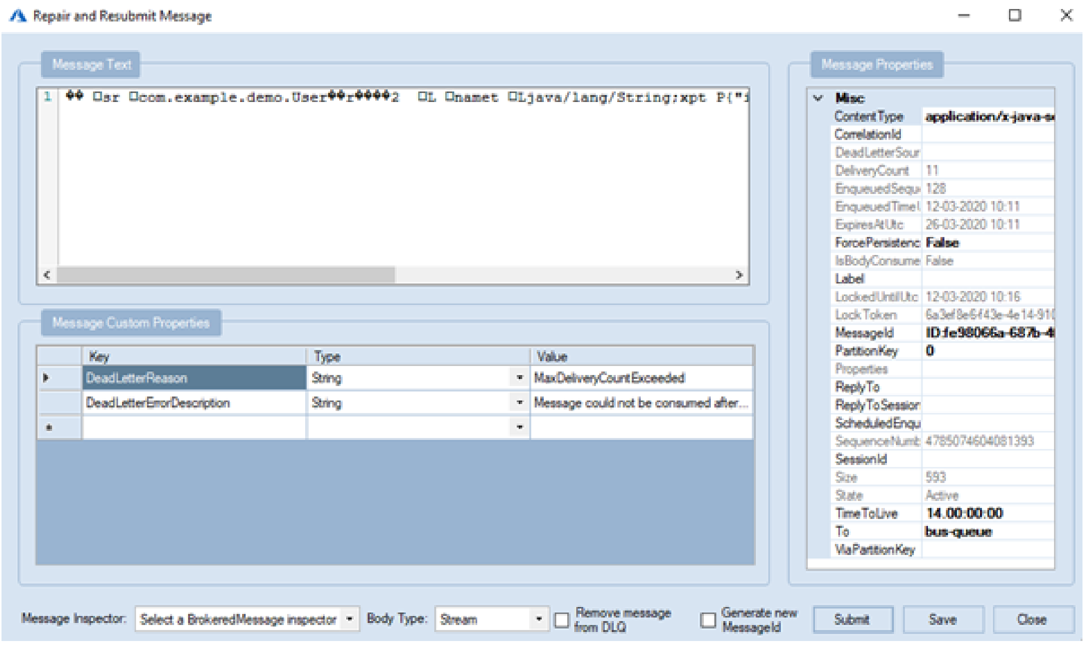
Here, in the below figure under **Message Properties** we can check the **Key , Type and Value** to repair and resubmit message.

>> Click on **Remove Message from DLQ** Check Button , to remove message from DLQ

>> Click on **Generate new MessageId** Check Button **,** to Generate new Message Id

>> Click on **Save**, to update Changes

>> Click on **Submit**, to accept the changes



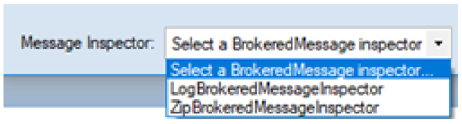
```
�� Dsr Dcom.example.demo.User��r����2  DL Dnamet DLjava/lang/String;xpt P{"i
```

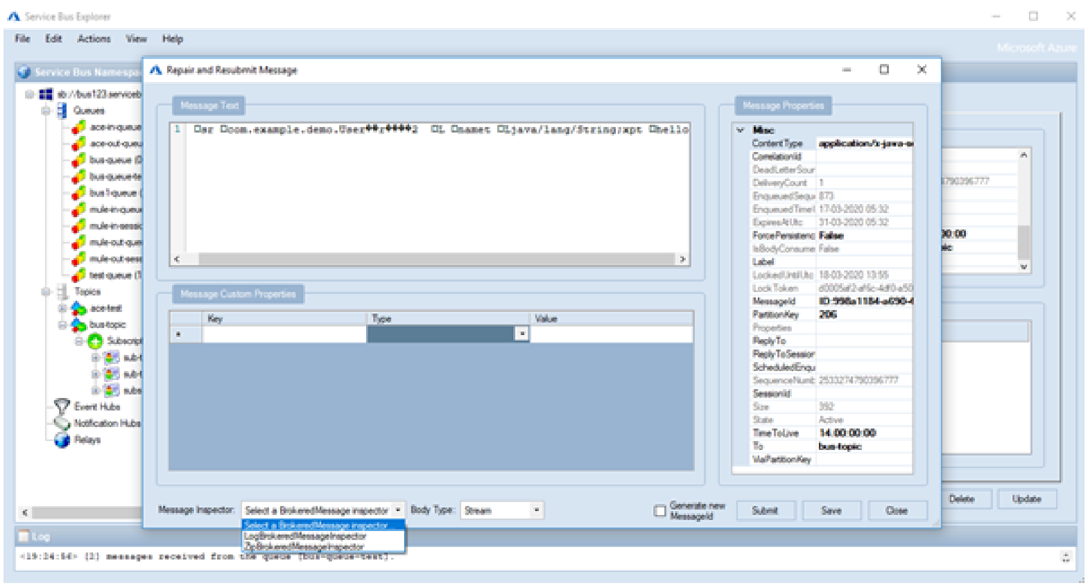
	Key	Type	Value
▸	DeadLetterReason	String	MaxDeliveryCountExceeded
	DeadLetterErrorDescription	String	Message could not be consumed after...
*			

Misc	
ContentType	application/x-java-se
CorrelationId	
DeadLetterSour	
DeliveryCount	11
EnqueuedSequ	128
EnqueuedTimeU	12-03-2020 10:11
ExpiresAtUtc	26-03-2020 10:11
ForcePersistenc	False
IsBodyConsume	False
Label	
LockedUntilUtc	12-03-2020 10:16
LockToken	6a3ef8e6f43e-4e14-910
MessageId	ID:fe98066a-687b-4
PartitionKey	0
Properties	
ReplyTo	
ReplyToSession	
ScheduledEnqu	
SequenceNumb	4785074604081393
SessionId	
Size	593
State	Active
TimeToLive	14.00:00:00
To	bus-queue
ViaPartitionKey	

Message Inspector: Select a BrokeredMessage inspector   Body Type: Stream   ☐ Remove message from DLQ   ☐ Generate new MessageId

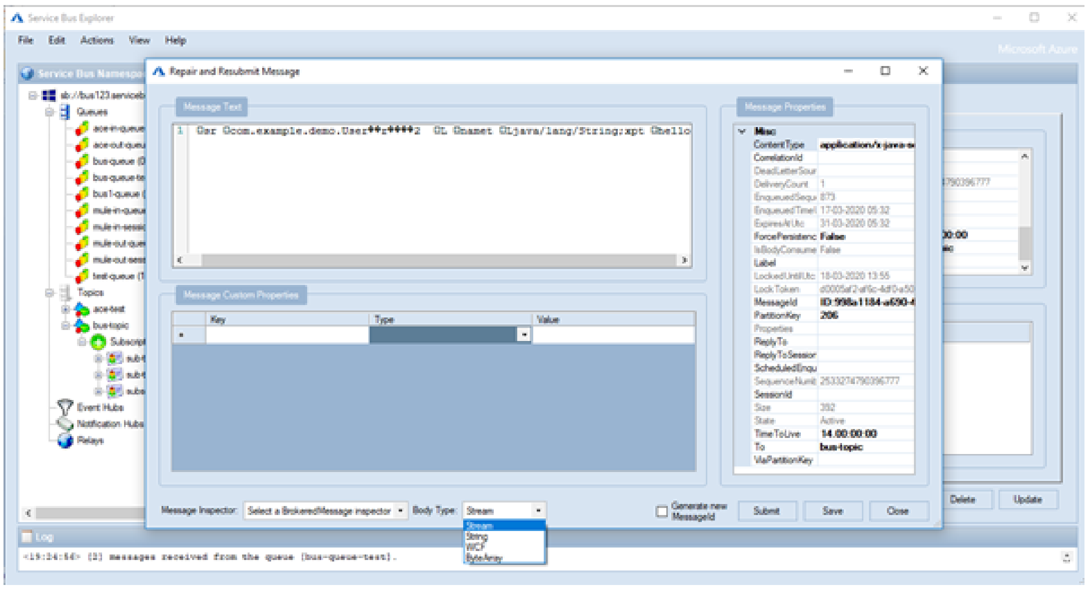
>> Choose one of the following from **Message Inspector** drop down menu


Message Inspector:
Select a BrokeredMessage inspector
Select a BrokeredMessage inspector...
LogBrokeredMessageInspector
ZipBrokeredMessageInspector


Service Bus Explorer
File Edit Actions View Help
Repair and Resubmit Message
Message Text
Message Custom Properties
Key Type Value
Message Properties
Message Inspector: Select a BrokeredMessage inspector
Body Type: Stream
Generate new MessageId
Submit Save Close
Delete Update
Select a BrokeredMessage inspector
LogBrokeredMessageInspector
ZipBrokeredMessageInspector
Log

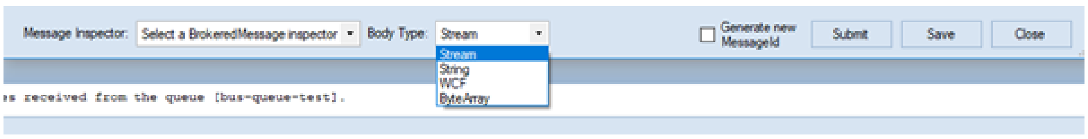
```
<19:24:54> [2] messages received from the queue [bus-queue-test].
```

>> Selecting **Body Type** for Message Text



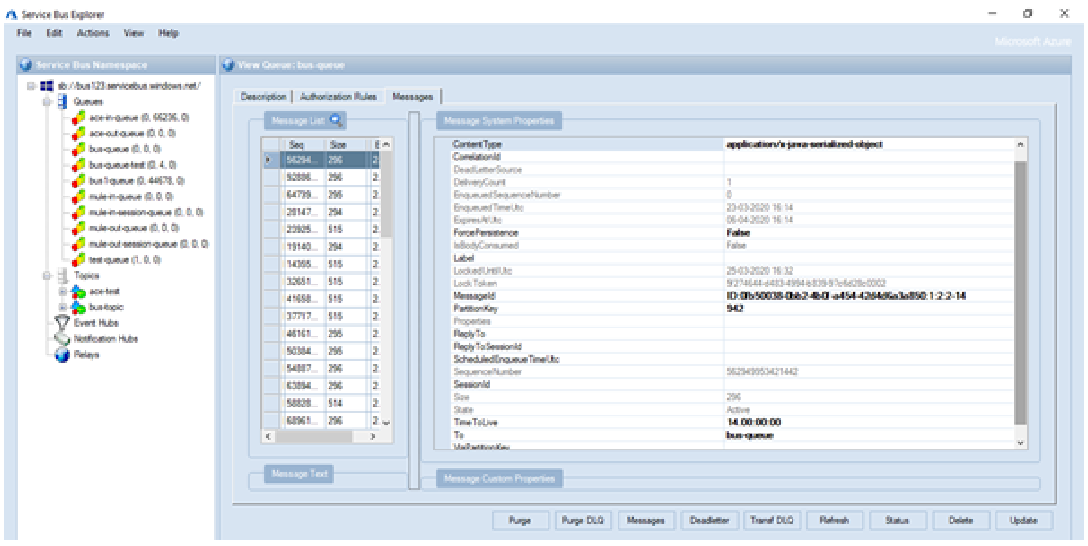
```
<19:24:54> [2] messages received from the queue [bus-queue-test].
```

>> Detail view of **Body Type** >> Select drop down menu to choose one of the following



```
s received from the queue [bus-queue-test].
```

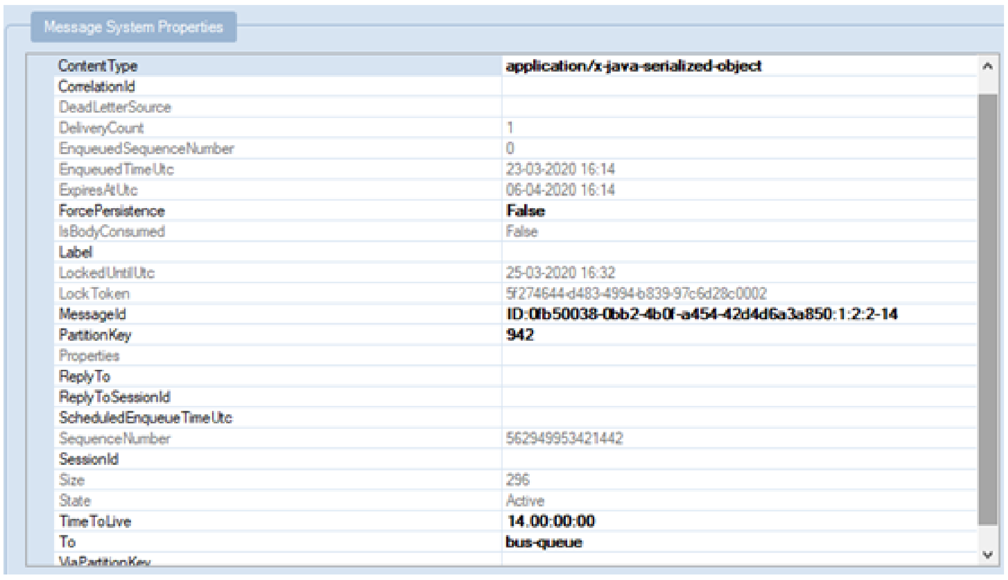
>> **Message System Properties**



- sb://bus123.servicebus.windows.net/
  - Queues
    - ace-in-queue (0, 66236, 0)
    - ace-out-queue (0, 0, 0)
    - bus-queue (0, 0, 0)
    - bus-queue-test (0, 4, 0)
    - bus1-queue (0, 44678, 0)
    - mule-in-queue (0, 0, 0)
    - mule-in-session-queue (0, 0, 0)
    - mule-out-queue (0, 0, 0)
    - mule-out-session-queue (0, 0, 0)
    - test-queue (1, 0, 0)
  - Topics
    - ace-test
    - bus-topic
  - Event Hubs
  - Notification Hubs
  - Relays

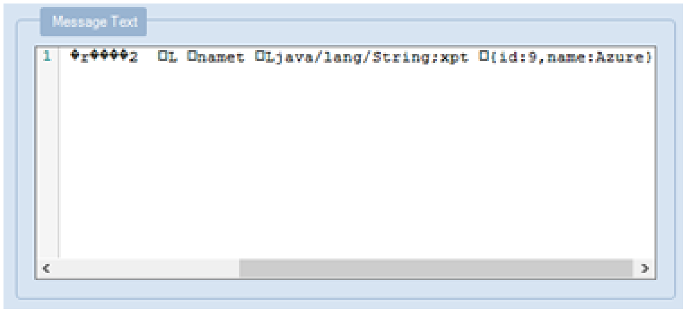
Seq	Size	E
56294...	296	2
52886...	296	2
64739...	295	2
28147...	294	2
23925...	515	2
19140...	294	2
14355...	515	2
32651...	515	2
41658...	515	2
37717...	515	2
46161...	295	2
50384...	295	2
54887...	296	2
63894...	296	2
58828...	514	2
68961...	296	2

>> Detail view of **Message System Properties**



ContentType	application/x-java-serialized-object
CorrelationId	
DeadLetterSource	
DeliveryCount	1
EnqueuedSequenceNumber	0
EnqueuedTimeUtc	23-03-2020 16:14
ExpiresAtUtc	06-04-2020 16:14
ForcePersistence	False
IsBodyConsumed	False
Label	
LockedUntilUtc	25-03-2020 16:32
LockToken	5f274644-d483-4994-b839-97c6d28c0002
MessageId	ID:0fb50038-0bb2-4b0f-a454-42d4d6a3a850:1:2:2-14
PartitionKey	942
Properties	
ReplyTo	
ReplyToSessionId	
ScheduledEnqueueTimeUtc	
SequenceNumber	562949953421442
SessionId	
Size	296
State	Active
TimeToLive	14.00:00:00
To	bus-queue
ViaPartitionKey	

>> **Message Text**



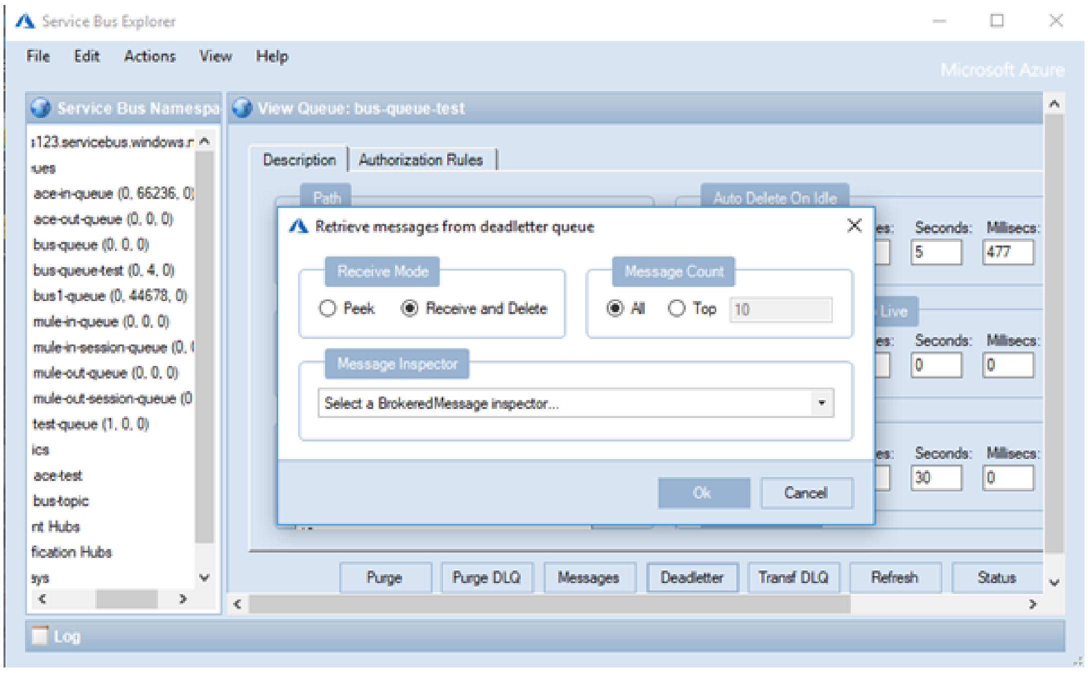
```
��r����2  L  Onamet  Ljava/lang/String;xpt  {id:9,name:Azure}
```

>> When you Click on **DeadLetter, Retrive messages from deadletter queue** pop up window menu will appear

>> Under **Receive Mode** >> Select radio button either **Peek** or **Receive and Delete**

>> Under **Message Count >>** Select radio button either **All** or **Top** (Count of messages in dialog box)

>> Under **Message Inspector >>** Select a **BrokeredMessage inspector** from drop down menu (Optional)

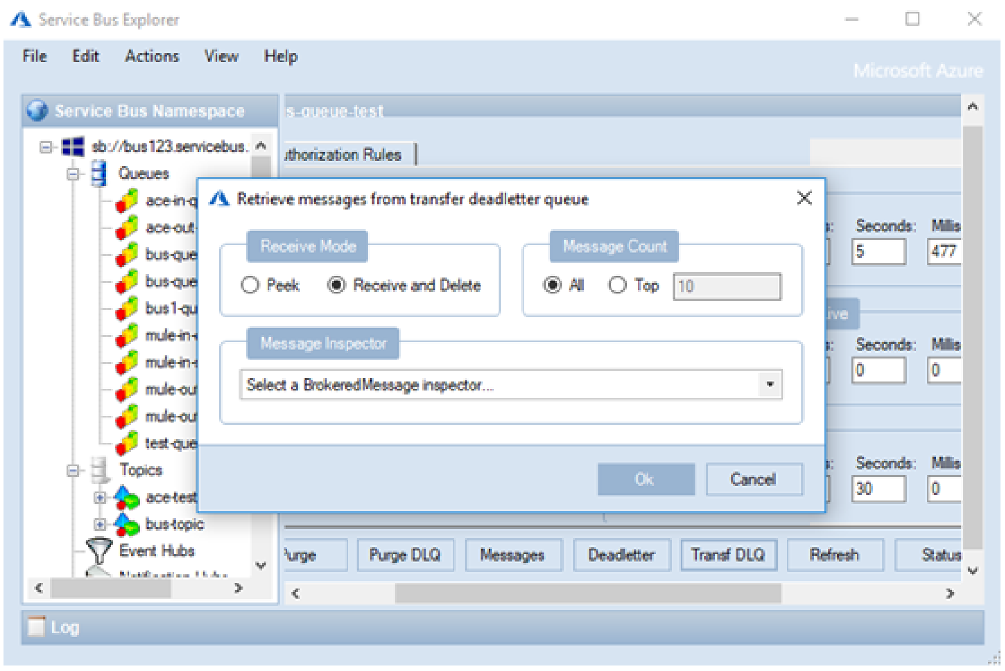


>> When you Click on **Transf DLQ, Retrive messages from Transf DLQ** pop up window menu will appear

>> Under **Receive Mode** >> Select radio button either **Peek** or **Receive and Delete**

>> Under **Message Count >>** Select radio button either **All** or **Top** (Count of messages in dialog box)

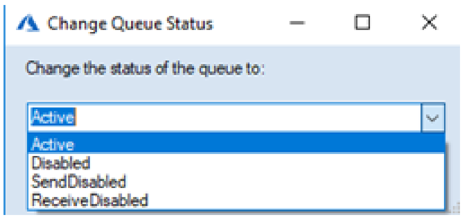
>> Under **Message Inspector >>** Select a **BrokeredMessage inspector** from drop down menu (Optional)

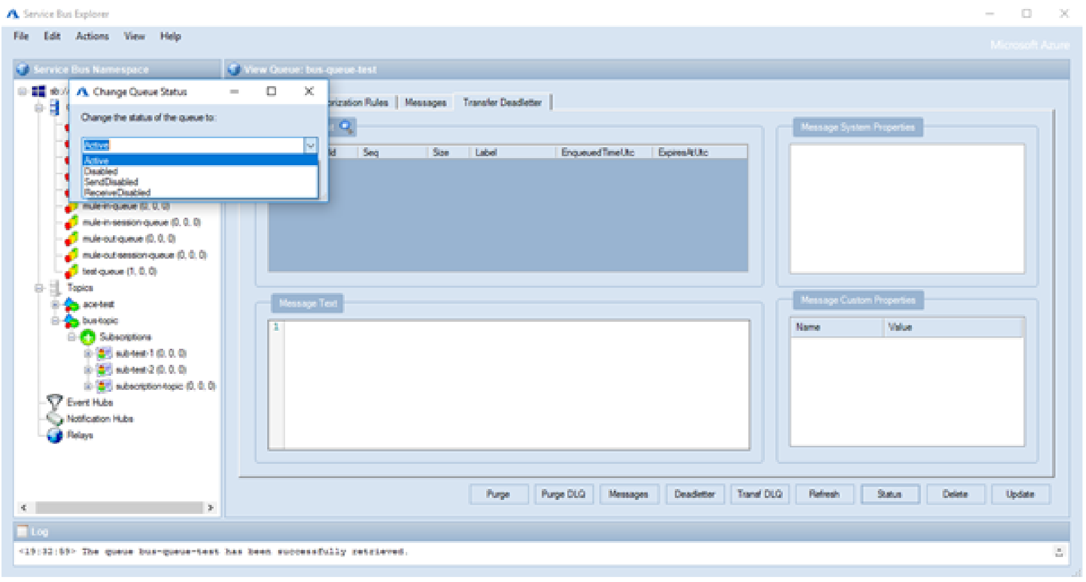


>> When you Click on **Refresh** Button, the resources will get refreshed

>> When you Click on **Status** Button, you can change status of Azure Service Bus queue, topic, or subscription to one of following:

- Active
- Disabled
- Receive Disabled
- Send Disabled



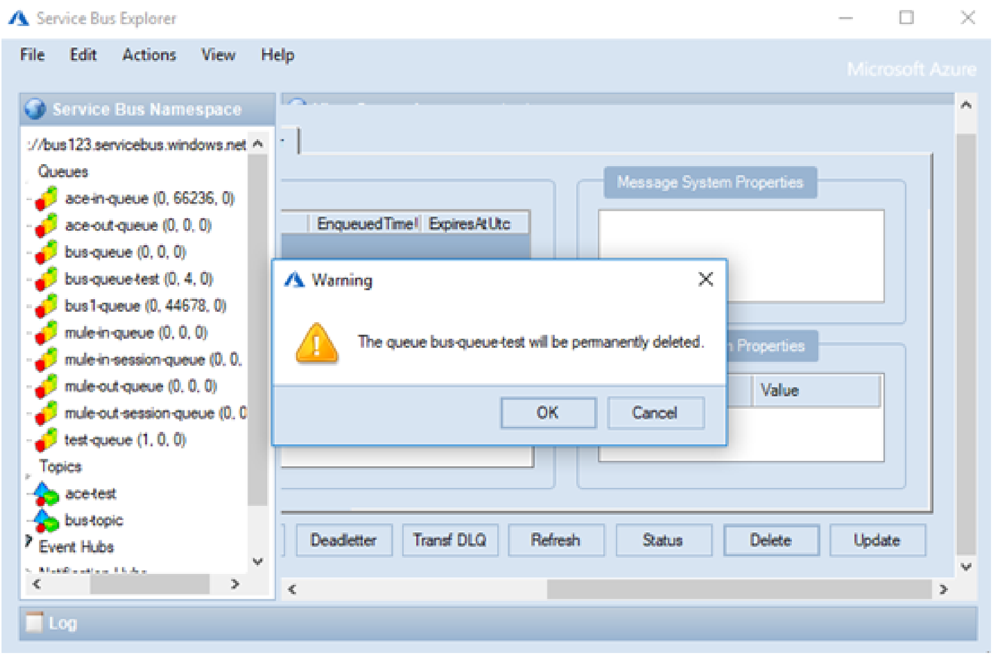


Change the status of the queue to:

Active
Active
Disabled
SendDisabled
ReceiveDisabled

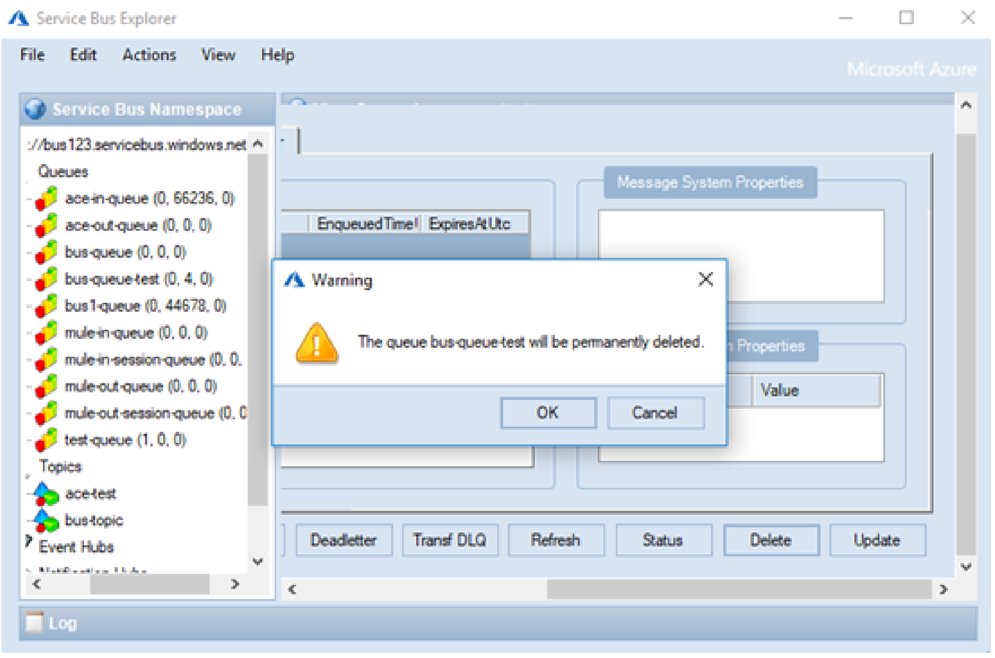
```
<19:32:59> The queue bus-queue-test has been successfully retrieved.
```

>> When You Click on **Delete** Button, the Selected resource will get deleted



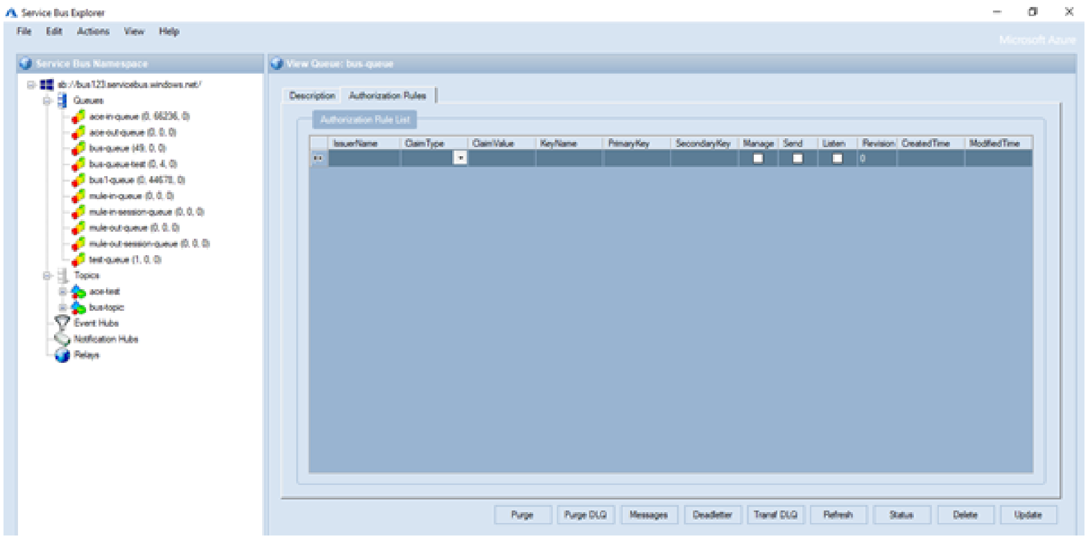
The queue bus-queue-test will be permanently deleted.

>> When you Click on **Update** Button, the updated changes will get updated

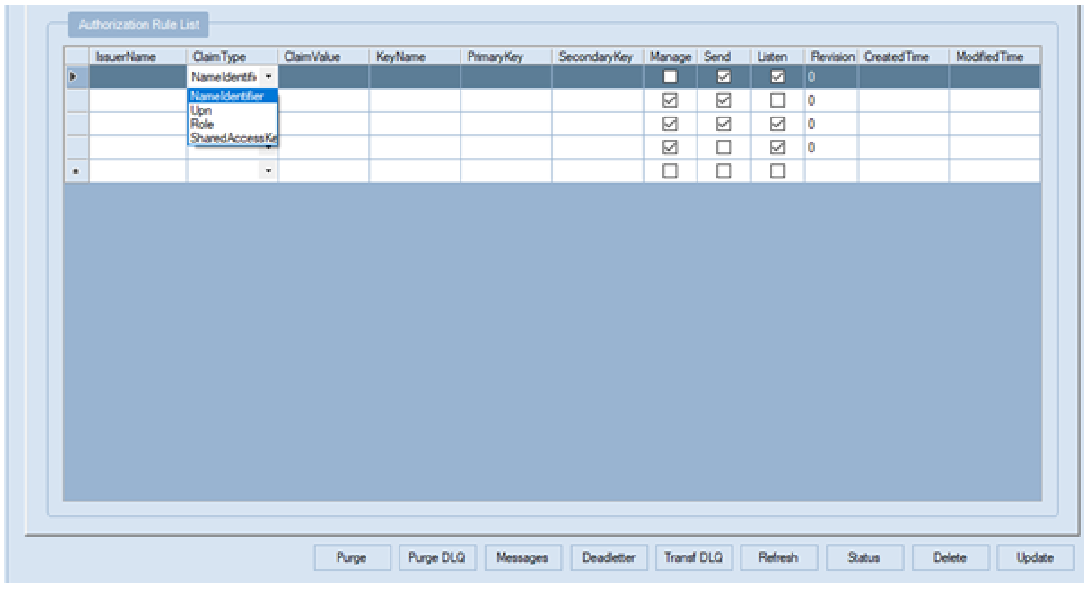


The queue bus-queue-test will be permanently deleted.

>> In **View Queue** >> **Authorization Rules**



>> **Authorization Rule List**



	IssuerName	ClaimType	ClaimValue	KeyName	PrimaryKey	SecondaryKey	Manage	Send	Listen	Revision	CreatedTime	ModifiedTime
▶		NameIdentifi					☐	☑	☑	0		
							☑	☑	☐	0		
							☑	☑	☑	0		
							☑	☐	☑	0		
*							☐	☐	☐			

Benefits for Developers:

With Service Bus Explorer we can dig into your queues and see which message went where, whether it was correctly formatted, with correct headers, etc. Multiple viewers are at your disposal: JSON, XML, WCF, Hex, .Net objects...

Develop and test sending and receiving applications or services separately -Create, store, analyze, edit, and resend test messages. That way we can test individual pieces of your solution, or even simulate third party messages going in or out.

Test performance and load handling of your app - Send thousands of messages for performance testing to see how system handles that load.

Benefits for Admins:

Troubleshoot, fix, and resend problematic messages - If something is wrong, Service Bus Explore is the fastest and easiest way to investigate and fix incorrect, poison, or failed/deadletter messages.

See status of your system at glance - Check if some of queues are growing too large, if there are messages in deadletter queues, etc. With refresh functionality see how entire system works in near real time.

Filter and sort messages to quickly find important data - Service Bus Explore allows us to quickly find messages we are interested in and extract our business data from within messages using XPath, JSON, or Regex.

Backup and restore messages - Backup messages to files, or move/copy them to another queue.

Manage "hidden" messages - Azure Service Bus has two special kinds of messages - deferred and scheduled, which your applications normally don't see. With Service Bus Explore we can check these messages, put them back in queue or reschedule them for another time.

DOCKERIZATION

Dockerizing an application is the process of converting an application to run within a Docker container.

The following are the steps to be followed for dockerization of a spring-boot application:
Firstly, run the spring-boot application by using the command $ mvn package.

When we run the application by using this command it produces a jar file in the target.

After generating the .jar file we make a **Dockerfile**
The contents of Dockerfile are as follows:

FROM
The FROM instruction initializes a new build stage and sets the Base Image for subsequent instructions. As such, a valid Dockerfile must start with a FROM instruction. The image can be any valid image – it is especially easy to start by pulling an image from the Public Repositories.

EXPOSE
The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime. You can specify whether the port listens on TCP or UDP, and the default is TCP if the protocol is not specified.

The EXPOSE instruction does not actually publish the port. It functions as a type of documentation between the person who builds the image and the person who runs the container, about which ports are intended to be published. To actually publish the port when running the container, use the -p flag on docker run to publish and map one or more ports, or the -P flag to publish all exposed ports and map them to high-order ports.

CMD
The CMD instruction has three forms:
CMD ["executable","param1","param2"] (exec form, this is the preferred form)
CMD ["param1","param2"] (as default parameters to ENTRYPOINT)
CMD command param1 param2 (shell form)

There can only be one CMD instruction in a Dockerfile. If you list more than one CMD then only the last CMD will take effect.

So the Dockerfile will be as follows:

```
------------------------------------------------------------------------------------------------------------
FROM java:8
WORKDIR /
ADD demo-0.0.1-SNAPSHOT.jar demo-0.0.1-SNAPSHOT.jar
EXPOSE 8080
CMD ["java", "-jar", "demo-0.0.1-SNAPSHOT.jar"]
------------------------------------------------------------------------------------------------------------
```

After making the Dockerfile, we run the "**docker build -t**" command, to run the docker.

References:

https://github.com/paolosalvatori/ServiceBusExplorer/blob/develop/docs/documentation.md

https://www.cogin.com/QueueExplorer/docs/introduction/general-connecting.php

Like Be the first to like this