Service Bus Listener activity

Last Modified on 08/24/2023 5:08 am EDT

Overview

The Service Bus Listener activity provides a powerful messaging mechanism to integrate your workflow with other applications or services. This activity integrates with Microsoft Azure Service Bus, which is a cloud-based messaging service used to connect and send information between applications, services and devices.

Azure Service Bus and Cora SeQuence

Use the Service Bus Listener activity to listen to messages and start or resume a workflow when data is transferred between different applications and services.

- Messages are transferred in a binary format that can contain JSON, XML, or plain text.
- The received data is saved in the workflow UACT table, in the *Message Body* column.
- Every message receives a unique *Message ID*.
- You can set up workflow expressions to extract the message data.

Before you add a Service Bus Listener activity to your workflow, you need to set up the service bus entity in Microsoft Azure.

The Service Bus Listener activity can currently work with two types of entities:

- **Queues**: one direction communication — one sender to one receiver. The message delivery follows a first in, first out (FIFO) method. That is, messages are received and processed in the same order in which they enter the queue. The service bus processes only one message at a time.
- **Topics and subscriptions**: one direction communication — one sender to several receivers, through subscriptions. Messages are sent to a topic and delivered to one or more subscriptions. The receiver does not communicate with the topic. Use topics if your activity requires sending messages to several receivers.

Use cases

You can use the Service Bus Listener activity to start or resume several types of processes. For example, you can set up a service bus activity for an inter-bank transfer that requires customer's approval based on specific conditions. The service bus sends a message to a queue and Cora SeQuence creates a process for the bank's representative to receive the customer's approval in writing.

Configure the Service Bus Listener activity

Prerequisites

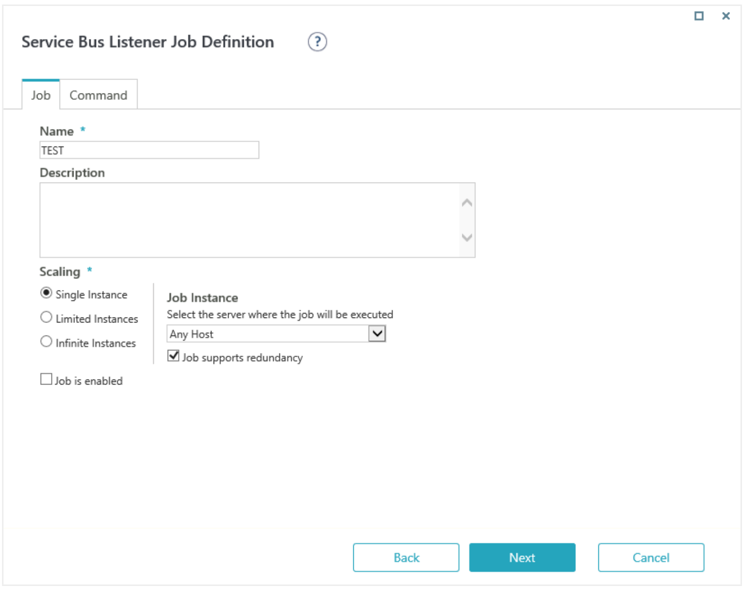
- Set up the Azure service bus namespace, which is a container for entities, such as queues and topics.
- Create a queue or topic, depending on your implementation requirement.
  For topics, you need to set up a subscription.
- Get the service bus Primary Connection String for your service bus namespace.

Procedure

1. To add a Service Bus Listener activity to your workflow, in the AppStudio, select **Integration>Service Bus**

Listener.

2. Double click **Service Bus Listener**.
3. On the Service Bus Listener Activity Properties screen, give a significant name to the activity, and then click **Next**.
4. On the Job tab, set the following:


Service Bus Listener Job Definition

Job | Command

Name *: TEST

Description

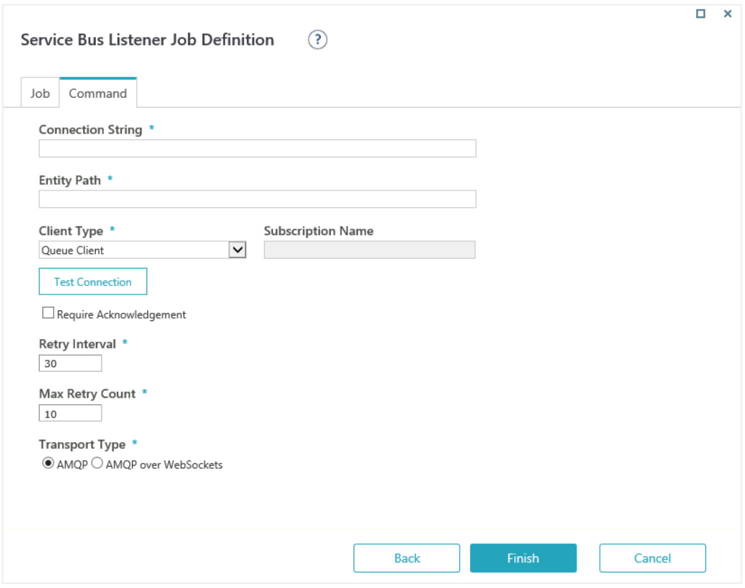
Scaling *
- (●) Single Instance
- ( ) Limited Instances
- ( ) Infinite Instances

Job Instance
Select the server where the job will be executed: Any Host
☑ Job supports redundancy

☐ Job is enabled

- **Name**: Enter a name to the actual job performed by the Service Bus Listener activity.
- **Scaling**: Set the required scaling settings. (Scaling is available only for **Cora SeQuence V9.3** and later versions.)
  For more details, see this article.
- **Job is enabled**: Select this option only after you complete the workflow, or if you want to run the job for testing purposes.

5. Click **Next**.
6. On the Command tab, set the following:


Service Bus Listener Job Definition

Job | Command

Connection String *

Entity Path *

Client Type *: Queue Client

Subscription Name

☐ Require Acknowledgement

Retry Interval *: 30

Max Retry Count *: 10

Transport Type *
(●) AMQP ( ) AMQP over WebSockets

- **Connection String**: Enter the primary connection string for your service bus namespace.
- **Entity Path**: Enter the name of the queue or topic, depending on what you configured for the service bus.
- **Client Type**: Do one of the following:
  - **Queue Client**: Select this option if the service bus uses a queue to transfer messages.

    - **Subscription Client**: Select this option, if the service bus uses a topic to transfer messages.
        - **Subscription Name**: Enter the name of the subscription that was set up for this topic.
- **Test Connection**: Verifies that the parameters you entered are valid, and Cora SeQuence can connect with the service bus.
- **Require Acknowledgement**: If you select this option, the message is kept in the queue until Cora SeQuence acknowledges that it successfully started or resumed the process. Only then the message is deleted.
- **Retry Interval**: Set the time (seconds) for the job to retry after failure.
  Maximum value is 999.
- **Max Retry Count:** The maximum number of times the system tries to rerun the job.
  Maximum value is 60.
- **Transport Type** (available only for Cora SeQuence V9.3 and later versions): Depending of the type of the Service Bus connection, select one of the options:
    - **AMQP**: Send messages using the Advanced Message Queuing Protocol (AMQP) over a TCP connection.
    - **AMQP over WebSockets**: Send messages using the Advanced Message Queuing Protocol (AMQP) over an HTTPS connection.

7. Click **Finish.**

Resume a workflow with the Service Bus Listener activity

Setting up the Service Bus Listener activity to resume a workflow requires additional configuration and integration with an external service.

After you create the Service Bus Listener activity, it waits for a message to arrive to a queue or topic configured for it. If the `Properties` property of the message contains the `JesActivityInstanceId` key with a value that matches an instance of a Service Bus Listener activity, the Job Execution Service picks up the message and resumes the execution of the workflow.

Procedure

1. In the App Studio:
    a. Place the Service Bus Listener activity anywhere in the workflow, except right after Start.
    b. Get the Service Bus Listener activity instance ID using an expression.
       Example: `{ServiceBusListener}.ActivityInstanceId`
2. In the external service:
    a. Send the Service Bus Listener activity instance ID and its value as a property named `JesActivityInstanceId` to the queue or topic configured in the activity.

Code sample

```
namespace ResumeAWorkflowByCallingAzureServiceBusListener
{
using Microsoft.ServiceBus.Messaging;
class Program
   {
static void Main(string[] args)
       {
long activityInstanceId = 3334;

           BrokeredMessage brokeredMessage = new BrokeredMessage();
           brokeredMessage.Properties.Add("JesActivityInstanceId", activityInstanceId);
       }
   }
}
```

NOTE
If the Service Bus Listener activity was already executed, depending on permissions, the activity may be executed again.

Limitations and important notes

- Cora SeQuence only checks for duplicate messages in the same workflow.
- You cannot have two workflows with the same Service Bus Listener activity job definition. If you copy a workflow, you need to redefine the Service Bus Listener job settings.
- Cora SeQuence does not support sessions.
- When you copy or create a new version of a workflow that contains the Service Bus Listener activity, you need to delete the Service Bus Listener activity in the new workflow, and then set it up again.

Want to learn by doing?
Check out this hands-on exercise.
Note that only internal users can access the link.