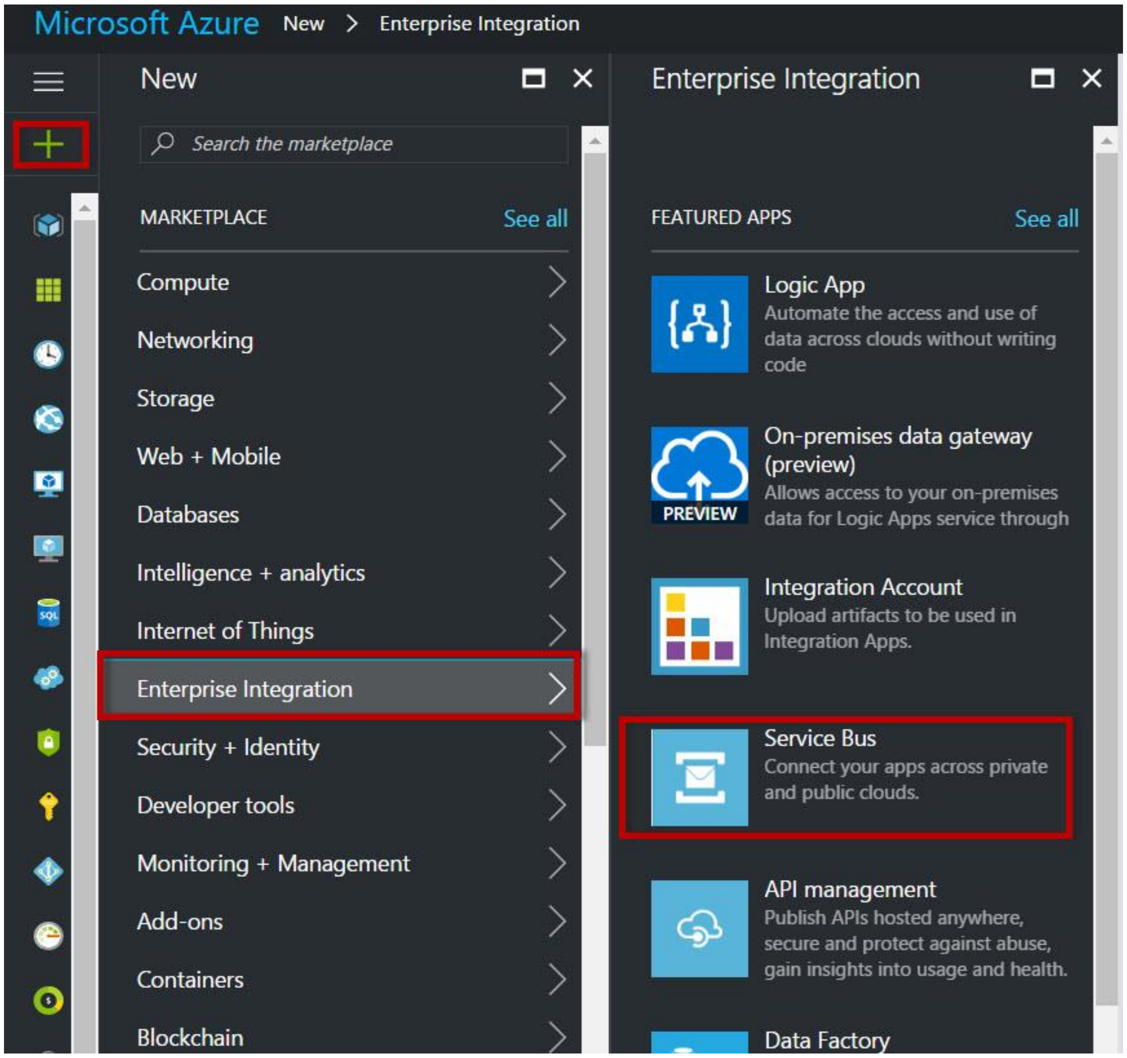
Azure Service Bus – Queue

Step 1: Navigate to Azure Portal.

Step 2: Click on +New -> Enterprise Integration -> Service Bus


Microsoft Azure New > Enterprise Integration

New

Search the marketplace

MARKETPLACE See all

Compute

Networking

Storage

Web + Mobile

Databases

Intelligence + analytics

Internet of Things

Enterprise Integration

Security + Identity

Developer tools

Monitoring + Management

Add-ons

Containers

Blockchain

Enterprise Integration

FEATURED APPS See all

Logic App
Automate the access and use of data across clouds without writing code

On-premises data gateway (preview)
Allows access to your on-premises data for Logic Apps service through

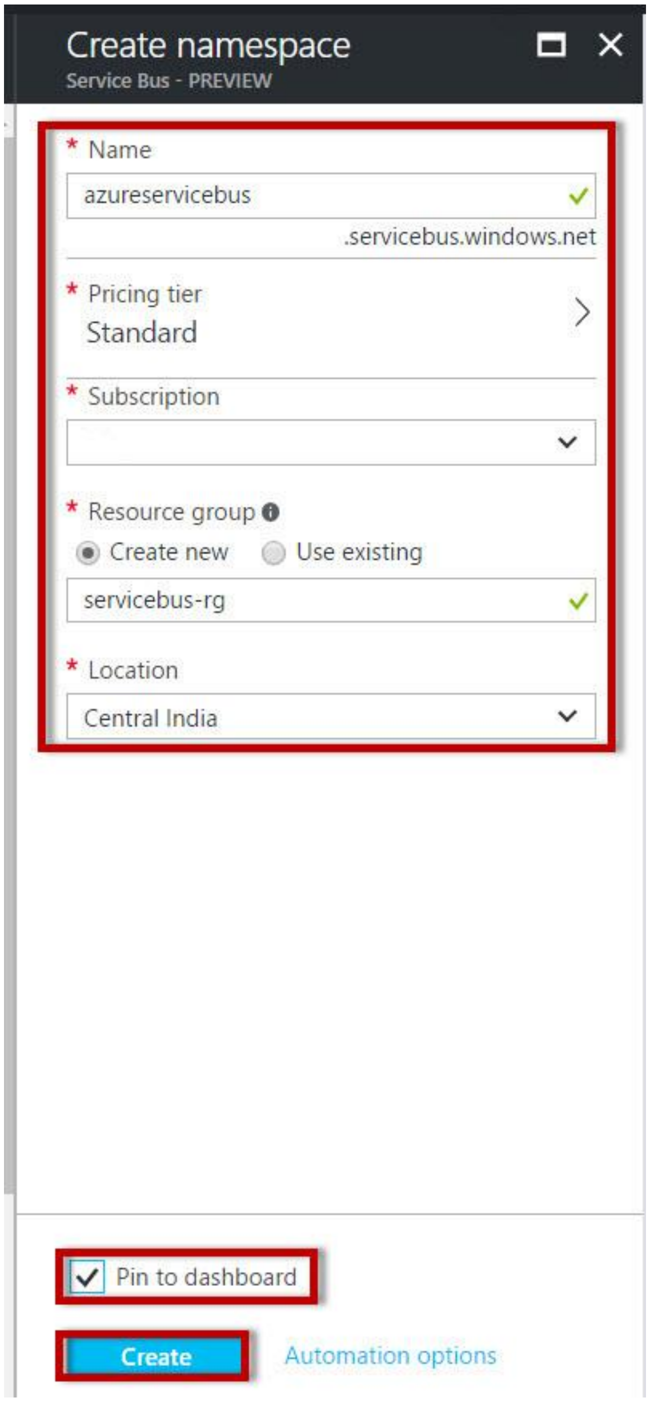
Integration Account
Upload artifacts to be used in Integration Apps.

Service Bus
Connect your apps across private and public clouds.

API management
Publish APIs hosted anywhere, secure and protect against abuse, gain insights into usage and health.

Data Factory

Step 3: Enter Service Bus Namespace Name, Choose Pricing Tier, Resource Group, Location


Create namespace
Service Bus - PREVIEW

* Name
azureservicebus ✓
.servicebus.windows.net

* Pricing tier
Standard

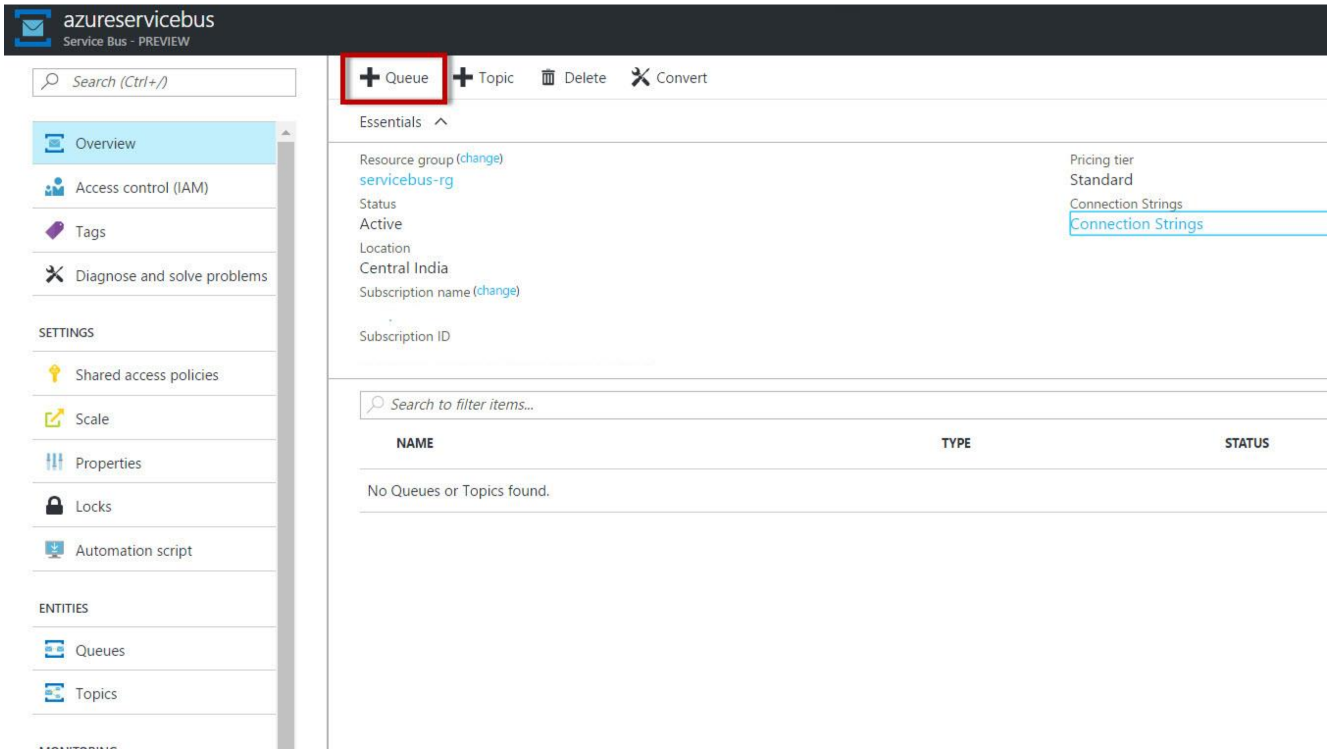
* Subscription

* Resource group
[✓ Create new] Use existing
servicebus-rg ✓

* Location
Central India

[✓] Pin to dashboard

Step 4: Click on **+ Queue** to create new Queue

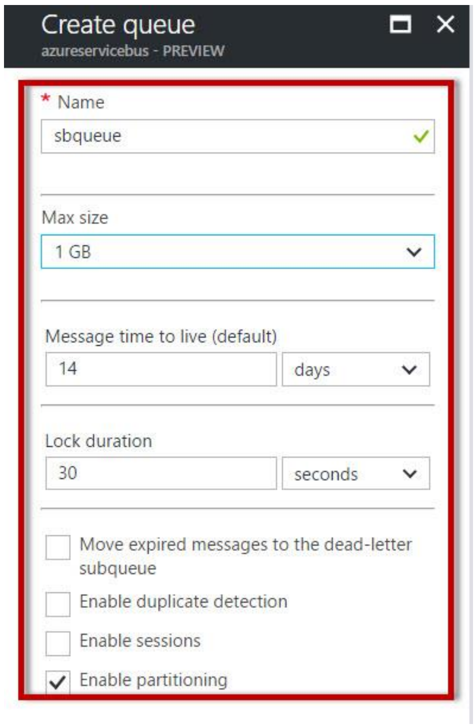


Resource group (change)
servicebus-rg
Status
Active
Location
Central India
Subscription name (change)
Subscription ID
Pricing tier
Standard
Connection Strings
Connection Strings

Search to filter items...

NAME	TYPE	STATUS
No Queues or Topics found.		

Step 5: Enter Queue Name, Max Size, Message time to live, Lock duration,etc.



* Name
sbqueue

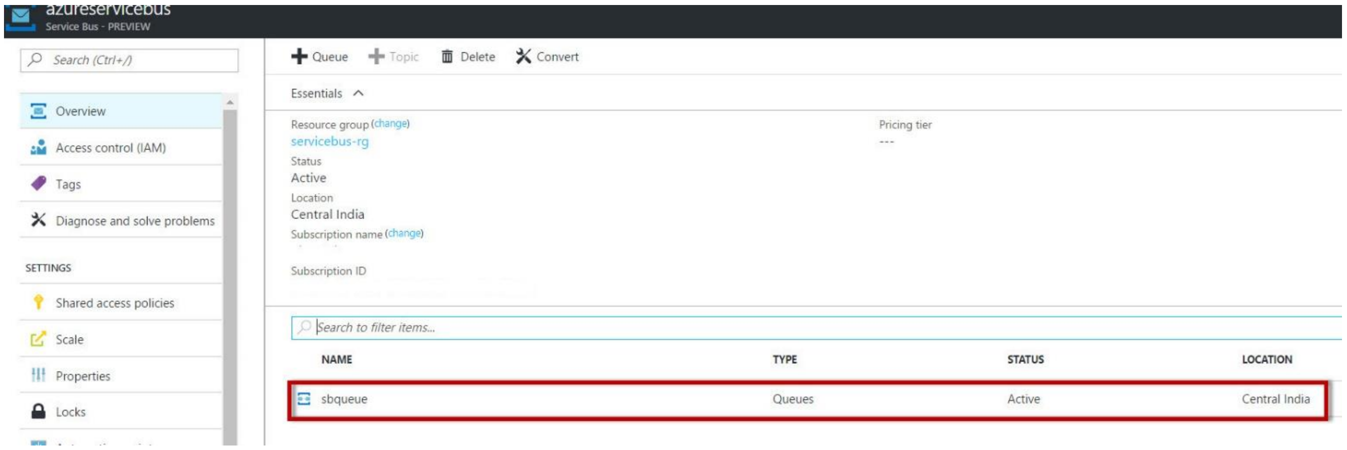
Max size
1 GB

Message time to live (default)
14 days

Lock duration
30 seconds

☐ Move expired messages to the dead-letter subqueue
☐ Enable duplicate detection
☐ Enable sessions
☑ Enable partitioning

Queue created successfully.

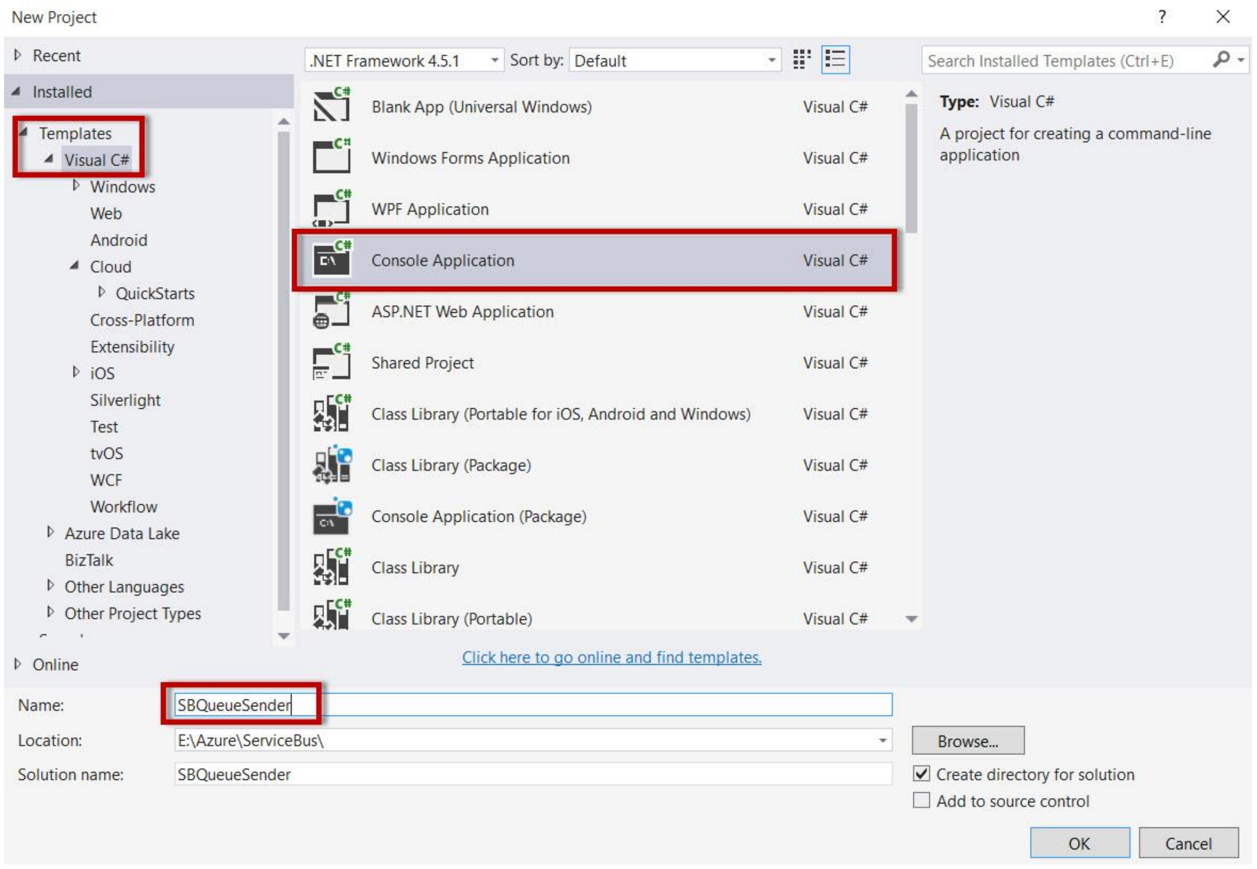


Resource group (change)
servicebus-rg
Status
Active
Location
Central India
Subscription name (change)
Subscription ID
Pricing tier

NAME	TYPE	STATUS	LOCATION
sbqueue	Queues	Active	Central India

Step 6: Start Visual Studio & Choose Visual C# Template.

Create New Console Application



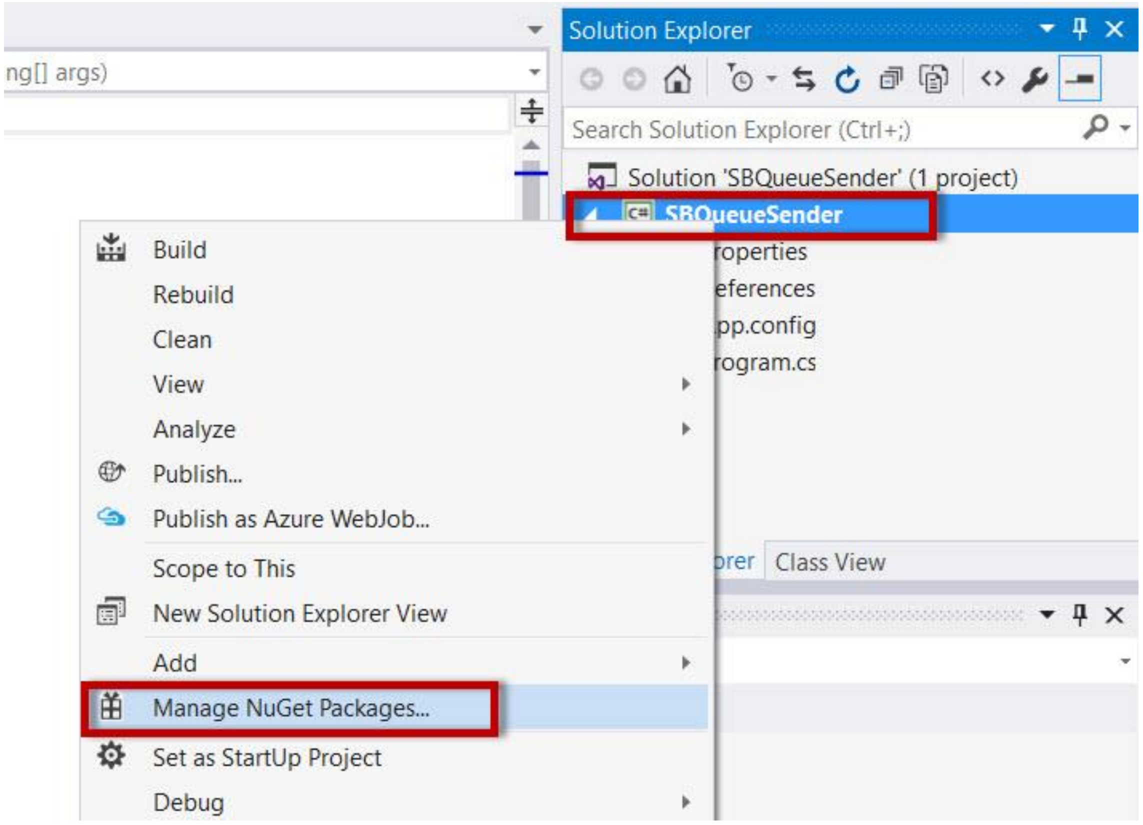
Blank App (Universal Windows)	Visual C#
Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
ASP.NET Web Application	Visual C#
Shared Project	Visual C#
Class Library (Portable for iOS, Android and Windows)	Visual C#
Class Library (Package)	Visual C#
Console Application (Package)	Visual C#
Class Library	Visual C#
Class Library (Portable)	Visual C#

Type: Visual C#

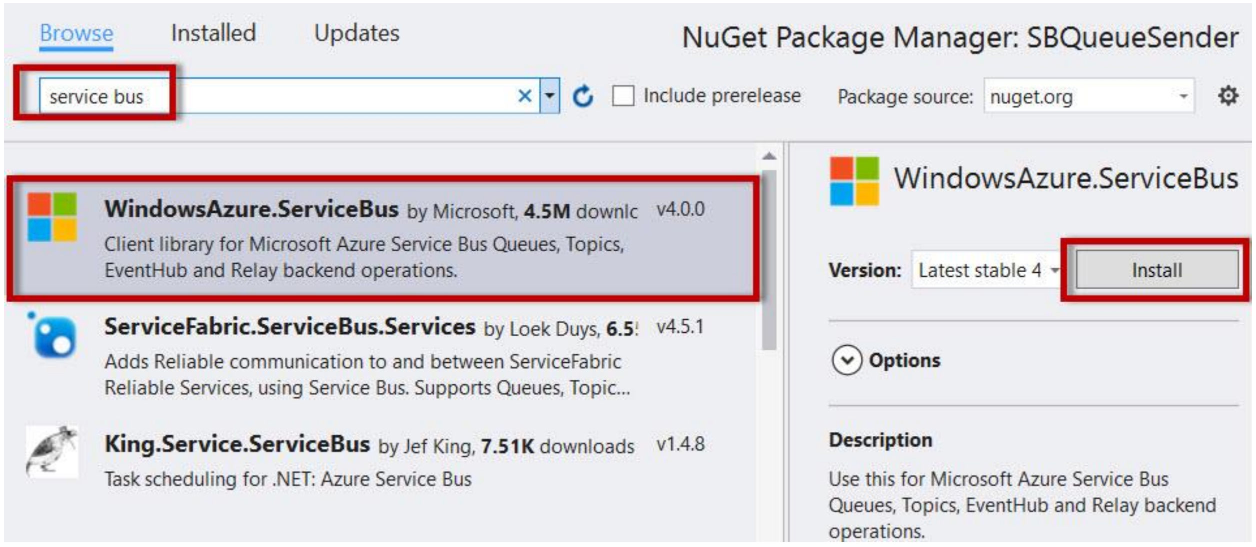
A project for creating a command-line application

Name: SBQueueSender
Location: E:\Azure\ServiceBus\
Solution name: SBQueueSender
[✓] Create directory for solution
[ ] Add to source control

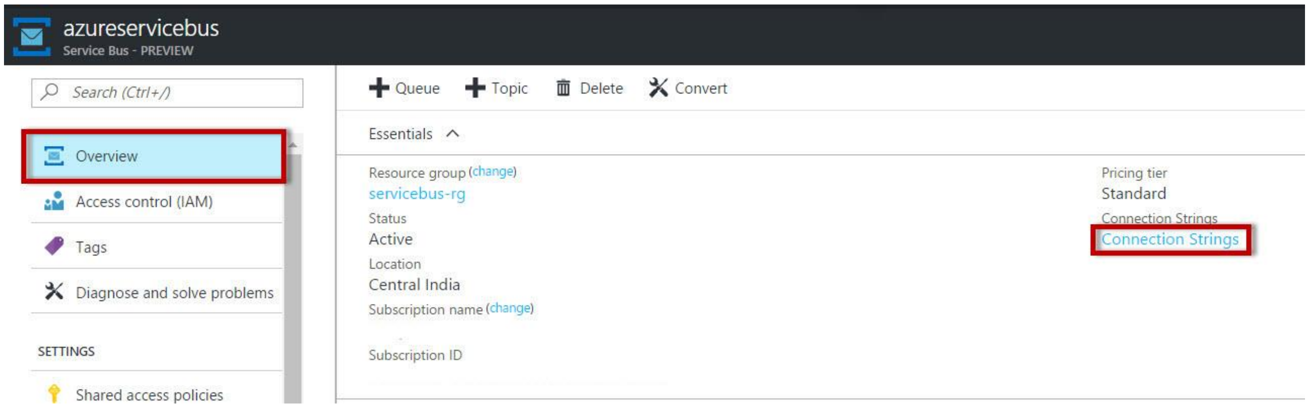
Step 7: Right click on Project Name & select “**Manage NuGet Packages…**”



Step 8: Search for “service bus” & install WindowsAzure.ServiceBus packages.

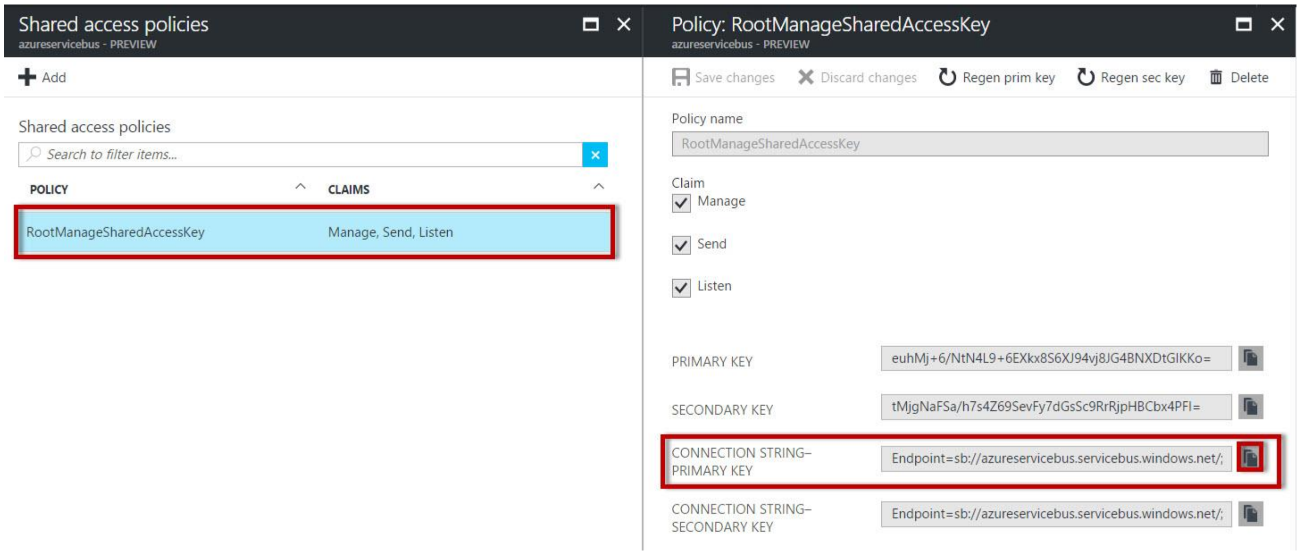

Browse
Installed
Updates
NuGet Package Manager: SBQueueSender
service bus
Include prerelease
Package source: nuget.org
WindowsAzure.ServiceBus by Microsoft, 4.5M downlc v4.0.0
Client library for Microsoft Azure Service Bus Queues, Topics, EventHub and Relay backend operations.
ServiceFabric.ServiceBus.Services by Loek Duys, 6.5! v4.5.1
Adds Reliable communication to and between ServiceFabric Reliable Services, using Service Bus. Supports Queues, Topic...
King.Service.ServiceBus by Jef King, 7.51K downloads v1.4.8
Task scheduling for .NET: Azure Service Bus
WindowsAzure.ServiceBus
Version: Latest stable 4
Install
Options
Description
Use this for Microsoft Azure Service Bus Queues, Topics, EventHub and Relay backend operations.

Step 9: Navigate to Azure Portal & Select **Connection String** option.



Resource group (change)
servicebus-rg
Status
Active
Location
Central India
Subscription name (change)
Subscription ID
Pricing tier
Standard
Connection Strings
Connection Strings

Click on Copy option & paste into **program.cs** file


Shared access policies
azureservicebus - PREVIEW
Add
Search to filter items...
Policy: RootManageSharedAccessKey
azureservicebus - PREVIEW
Save changes
Discard changes
Regen prim key
Regen sec key
Delete

Shared access policies

POLICY	CLAIMS
RootManageSharedAccessKey	Manage, Send, Listen

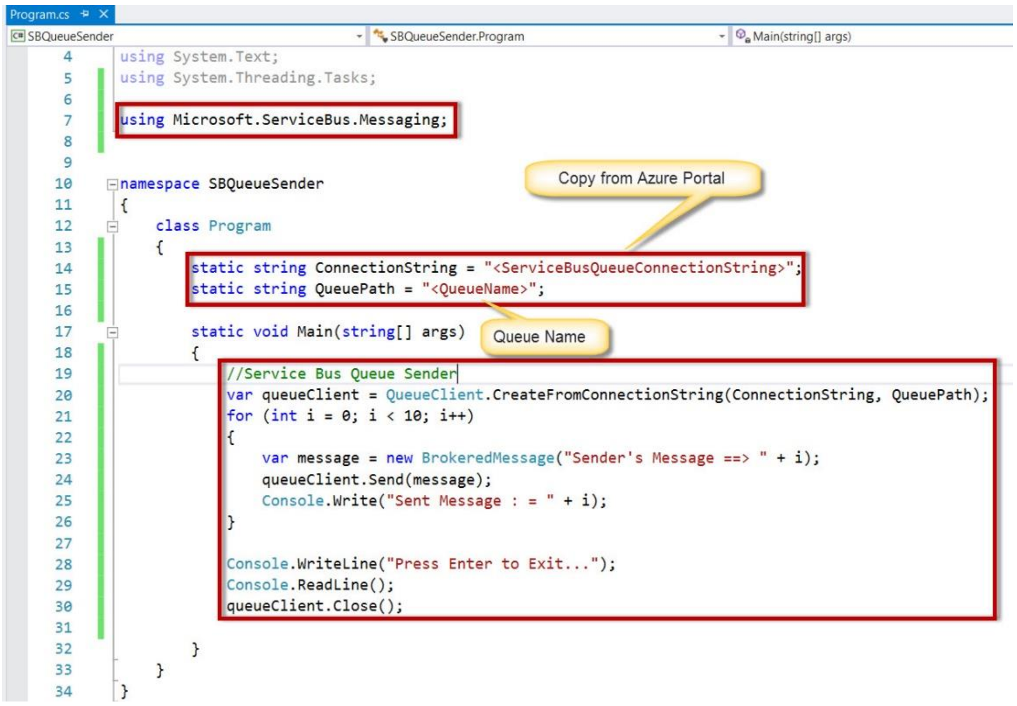
Policy name
RootManageSharedAccessKey

Claim
✓ Manage
✓ Send
✓ Listen

PRIMARY KEY	euhMj+6/NtN4L9+6EXkx8S6XJ94vj8JG4BNXDtGIKKo=
SECONDARY KEY	tMjgNaFSa/h7s4Z69SevFy7dGsSc9RrRjpHBCbx4PFI=
CONNECTION STRING–PRIMARY KEY	Endpoint=sb://azureservicebus.servicebus.windows.net/;
CONNECTION STRING–SECONDARY KEY	Endpoint=sb://azureservicebus.servicebus.windows.net/;

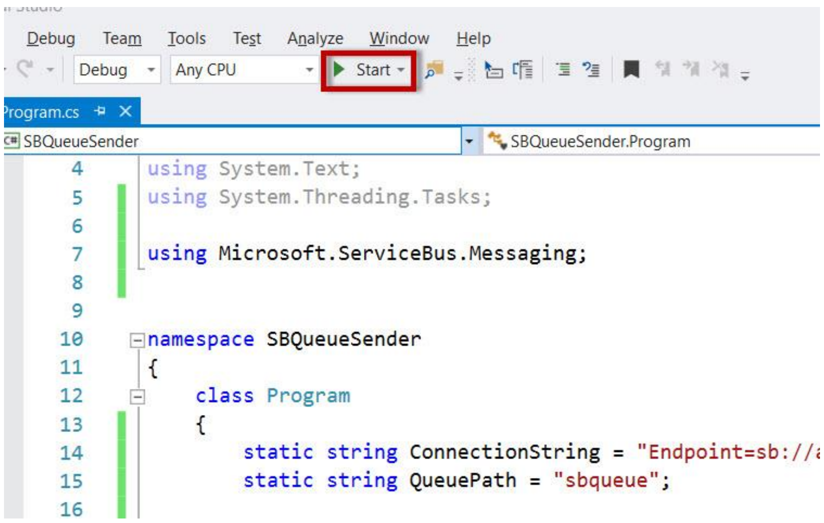
Step 10: Add Service Bus Namespace

```
using Microsoft.ServiceBus.Messaging;


namespace SBQueueSender
{
    class Program
    {
        static string ConnectionString =
"Endpoint=sb://azureservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageShar
edAccessKey;SharedAccessKey=<copy-servicebus-sharedkey-from-azure-portal>";
        static string QueuePath = "sbqueue";

        static void Main(string[] args)
        {
            //Service Bus Queue Sender
            var queueClient = QueueClient.CreateFromConnectionString(ConnectionString,
QueuePath);
            for (int i = 0; i < 10; i++)
            {
                var message = new BrokeredMessage("Sender's Message ==> " + i);


//      message.SessionId = “test”;


                queueClient.Send(message);
                Console.Write("\nSent Message : = " + i );
            }

            Console.WriteLine("Press Enter to Exit...");
            Console.ReadLine();
            queueClient.Close();

        }
    }
}
```

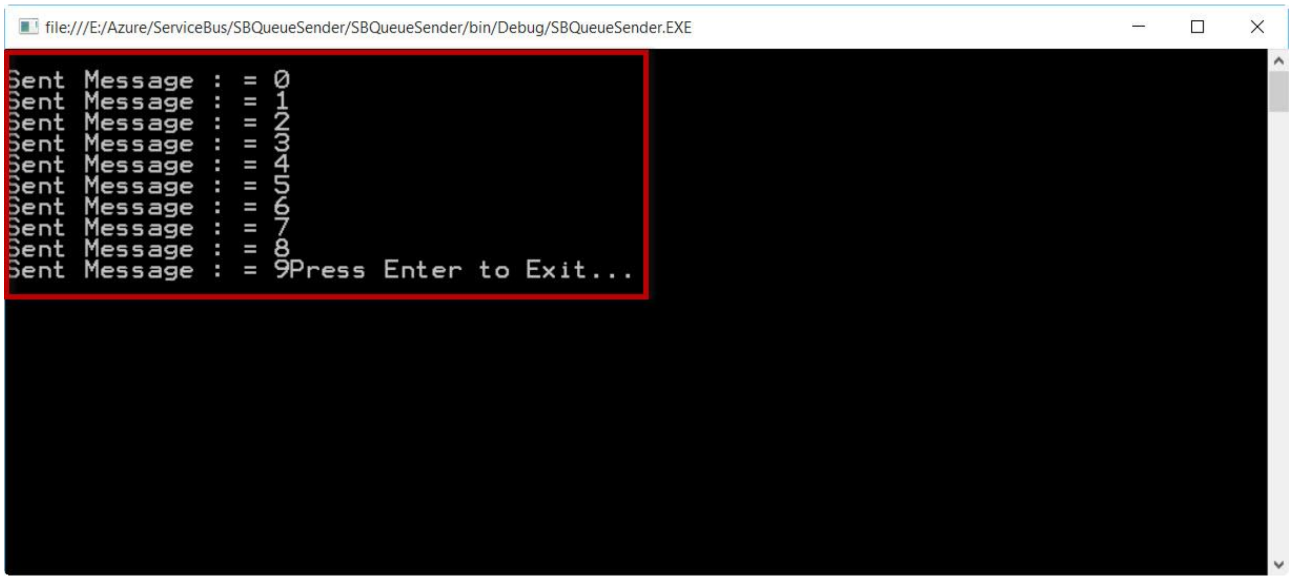


```
using System.Text;
using System.Threading.Tasks;

using Microsoft.ServiceBus.Messaging;


namespace SBQueueSender
{
    class Program
    {
        static string ConnectionString = "<ServiceBusQueueConnectionString>";
        static string QueuePath = "<QueueName>";

        static void Main(string[] args)
        {
            //Service Bus Queue Sender
            var queueClient = QueueClient.CreateFromConnectionString(ConnectionString, QueuePath);
            for (int i = 0; i < 10; i++)
            {
                var message = new BrokeredMessage("Sender's Message ==> " + i);
                queueClient.Send(message);
                Console.Write("Sent Message : = " + i);
            }

            Console.WriteLine("Press Enter to Exit...");
            Console.ReadLine();
            queueClient.Close();

        }
    }
}
```

Step 11: Now run the Sender program

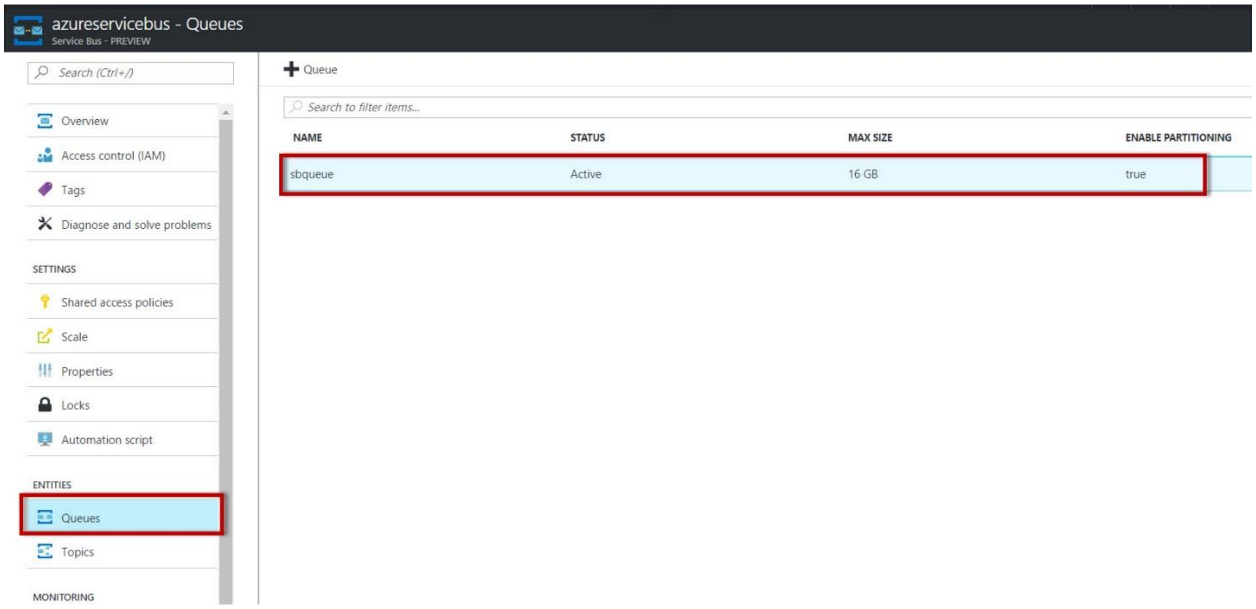


```
using System.Text;
using System.Threading.Tasks;

using Microsoft.ServiceBus.Messaging;


namespace SBQueueSender
{
    class Program
    {
        static string ConnectionString = "Endpoint=sb://
        static string QueuePath = "sbqueue";

```

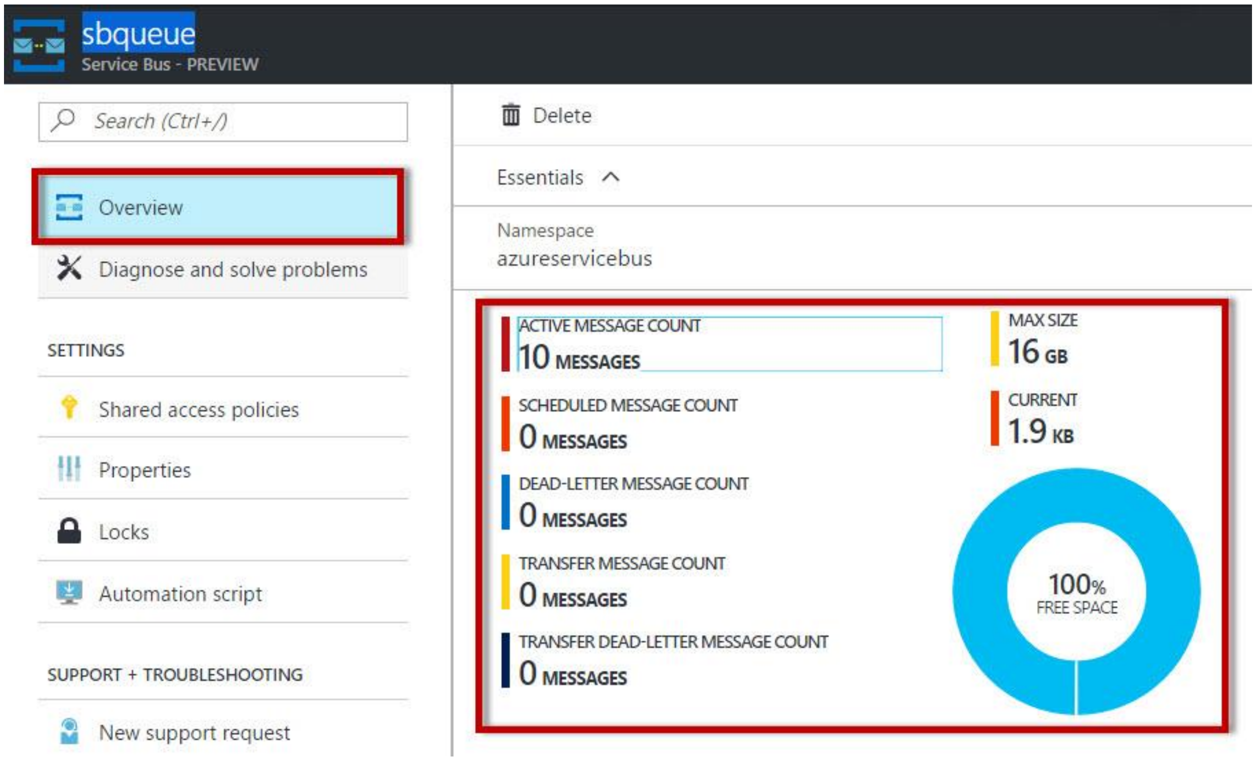

file:///E:/Azure/ServiceBus/SBQueueSender/SBQueueSender/bin/Debug/SBQueueSender.EXE

```
Sent Message : = 0
Sent Message : = 1
Sent Message : = 2
Sent Message : = 3
Sent Message : = 4
Sent Message : = 5
Sent Message : = 6
Sent Message : = 7
Sent Message : = 8
Sent Message : = 9Press Enter to Exit...
```

Again Navigate to Service Bus Queue option



NAME	STATUS	MAX SIZE	ENABLE PARTITIONING
sbqueue	Active	16 GB	true



Namespace
azureservicebus

ACTIVE MESSAGE COUNT
10 MESSAGES

SCHEDULED MESSAGE COUNT
0 MESSAGES

DEAD-LETTER MESSAGE COUNT
0 MESSAGES

TRANSFER MESSAGE COUNT
0 MESSAGES

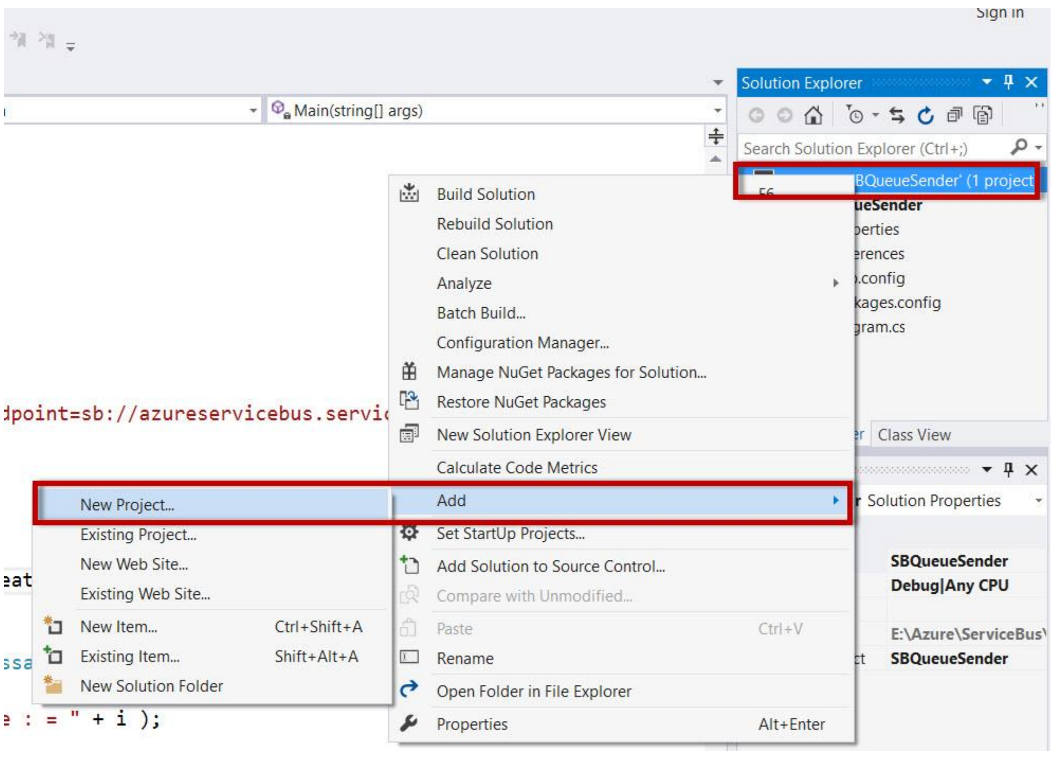
TRANSFER DEAD-LETTER MESSAGE COUNT
0 MESSAGES

MAX SIZE
16 GB

CURRENT
1.9 KB

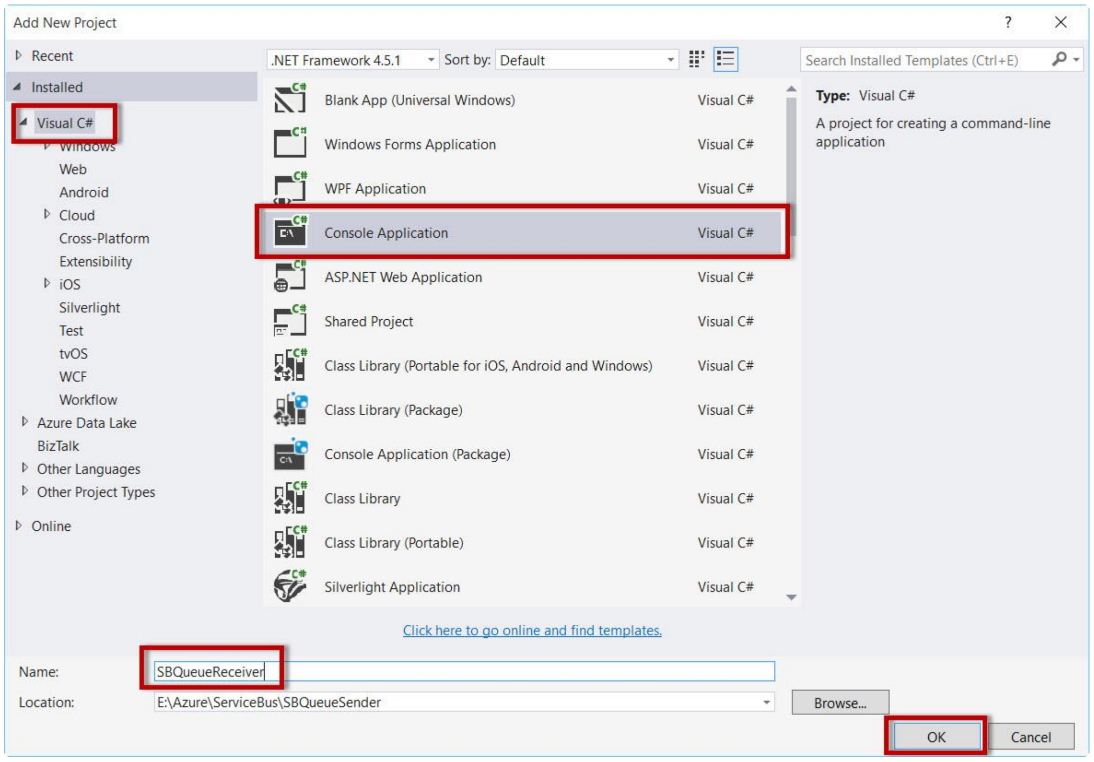
100%
FREE SPACE

Step 12: Create one more project for Receiver using Visual Studio.



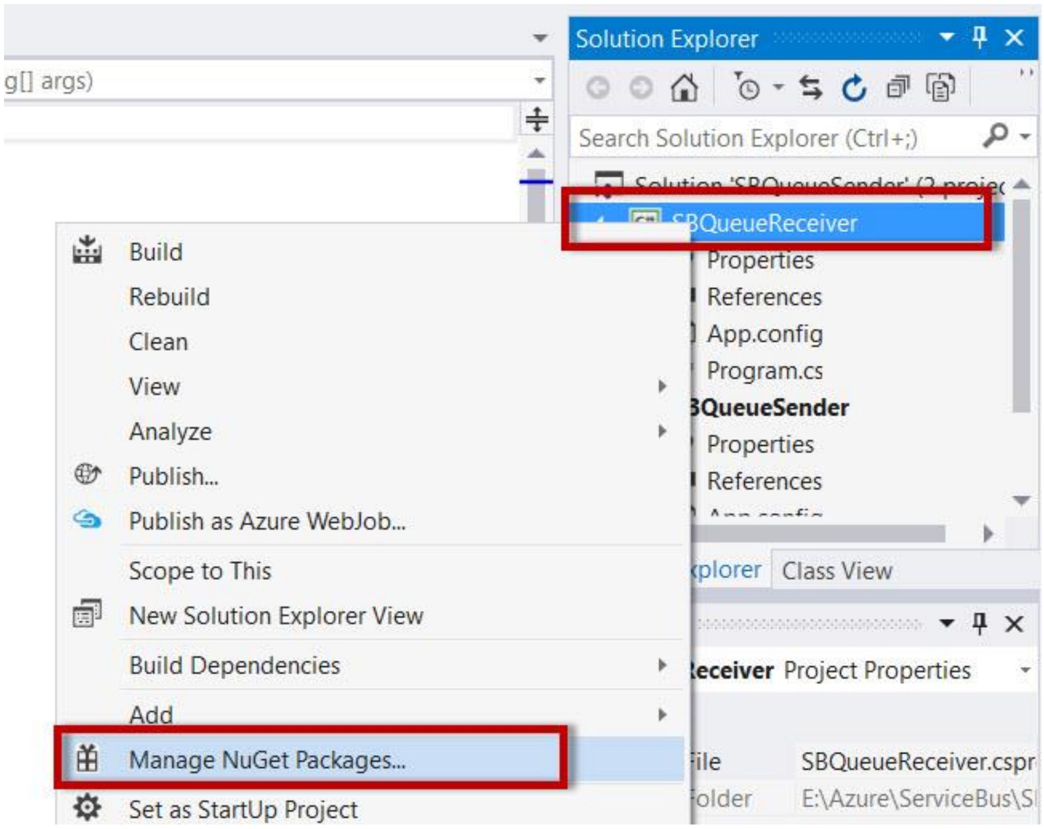
```
dpoint=sb://azureservicebus.servi
e : = " + i );
```

Create Console Application.

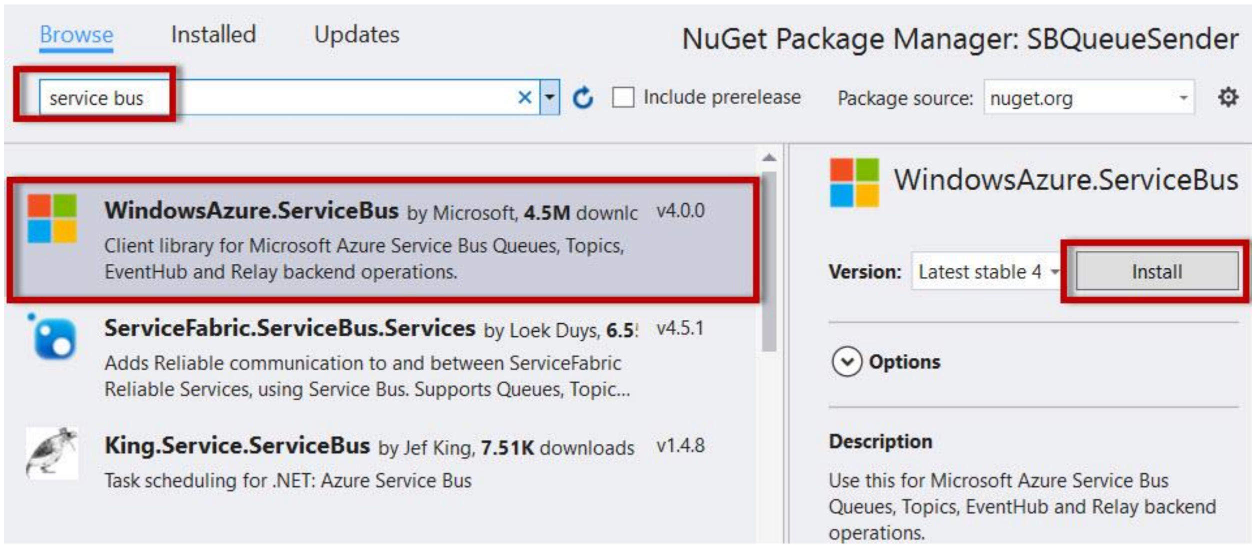

Add New Project
Recent
Installed
Visual C#
Windows
Web
Android
Cloud
Cross-Platform
Extensibility
iOS
Silverlight
Test
tvOS
WCF
Workflow
Azure Data Lake
BizTalk
Other Languages
Other Project Types
Online
.NET Framework 4.5.1
Sort by: Default
Search Installed Templates (Ctrl+E)
Blank App (Universal Windows) Visual C#
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
ASP.NET Web Application Visual C#
Shared Project Visual C#
Class Library (Portable for iOS, Android and Windows) Visual C#
Class Library (Package) Visual C#
Console Application (Package) Visual C#
Class Library Visual C#
Class Library (Portable) Visual C#
Silverlight Application Visual C#
Type: Visual C#
A project for creating a command-line application
Click here to go online and find templates.
Browse...
OK
Cancel

Name: SBQueueReceiver
Location: E:\Azure\ServiceBus\SBQueueSender

repeat the same steps to install NuGet Packages of Azure Service Bus.



Search for “service bus” & install NuGet Packages.



WindowsAzure.ServiceBus by Microsoft, 4.5M downlc v4.0.0
Client library for Microsoft Azure Service Bus Queues, Topics, EventHub and Relay backend operations.

ServiceFabric.ServiceBus.Services by Loek Duys, 6.5! v4.5.1
Adds Reliable communication to and between ServiceFabric Reliable Services, using Service Bus. Supports Queues, Topic...

King.Service.ServiceBus by Jef King, 7.51K downloads v1.4.8
Task scheduling for .NET: Azure Service Bus

WindowsAzure.ServiceBus

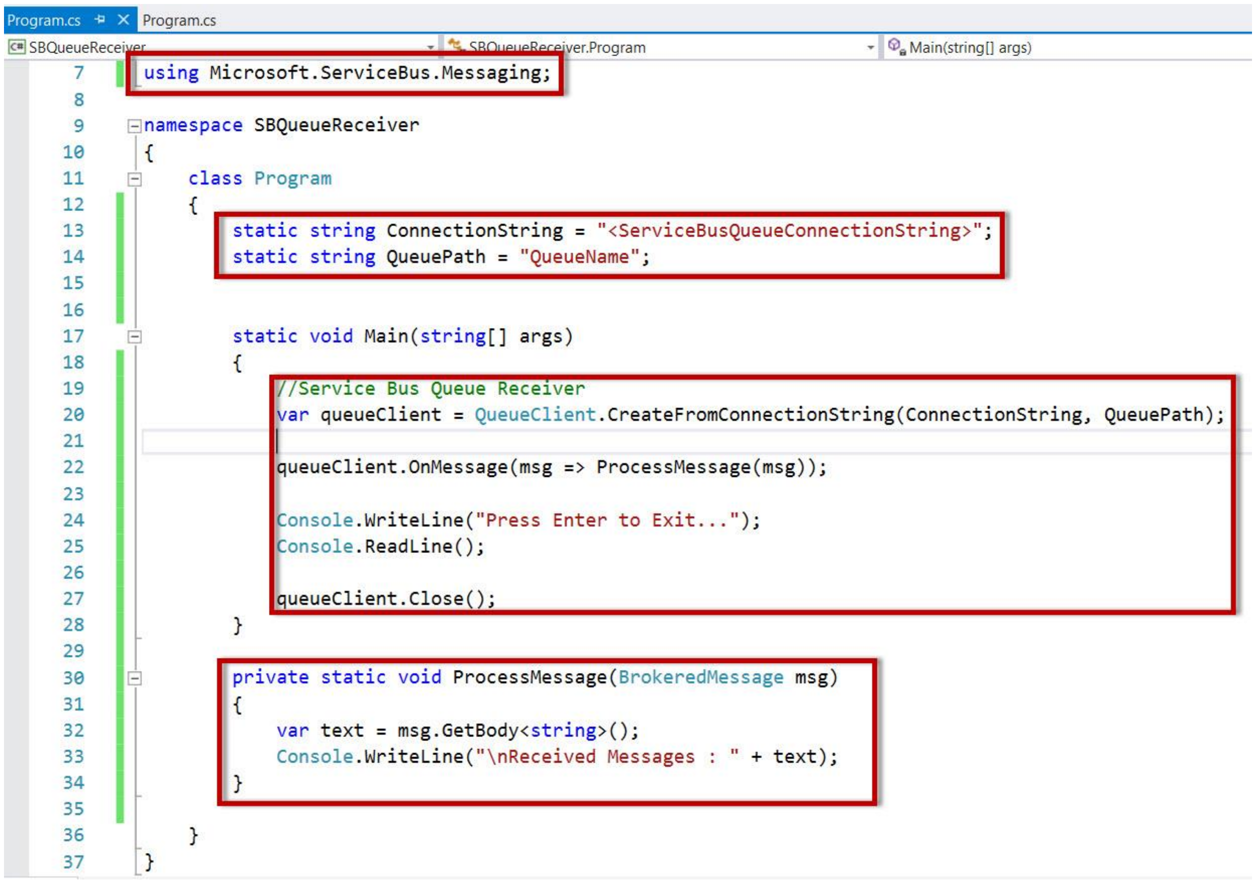
Use this for Microsoft Azure Service Bus Queues, Topics, EventHub and Relay backend operations.

Step 13:

Add Service Bus Namespace

using Microsoft.ServiceBus.Messaging;

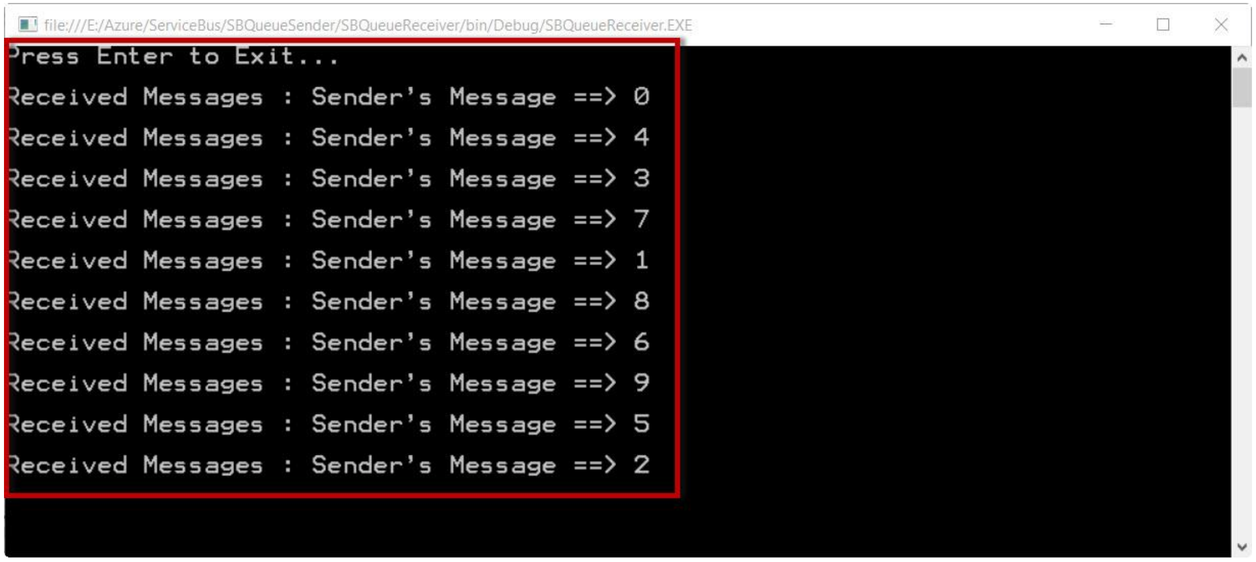
`Get ConnectionString & QueueName from Azure Portal or copy from Sender’s Project.`



```
using Microsoft.ServiceBus.Messaging;

namespace SBQueueReceiver
{
    class Program
    {
        static string ConnectionString = "<ServiceBusQueueConnectionString>";
        static string QueuePath = "QueueName";


        static void Main(string[] args)
        {
            //Service Bus Queue Receiver
            var queueClient = QueueClient.CreateFromConnectionString(ConnectionString, QueuePath);

            queueClient.OnMessage(msg => ProcessMessage(msg));

            Console.WriteLine("Press Enter to Exit...");
            Console.ReadLine();

            queueClient.Close();
        }

        private static void ProcessMessage(BrokeredMessage msg)
        {
            var text = msg.GetBody<string>();
            Console.WriteLine("\nReceived Messages : " + text);
        }

    }
}
```

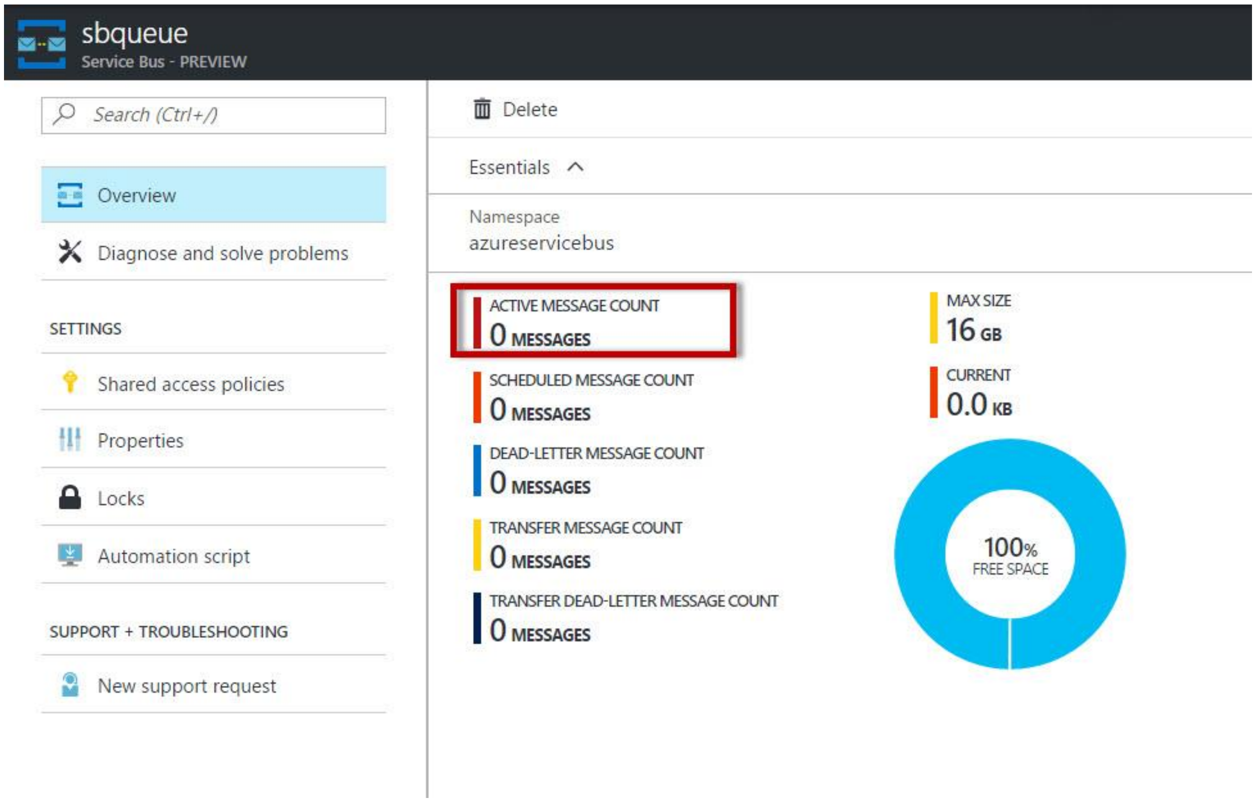
```
namespace SBQueueReceiver
{
    class Program
    {
        static string ConnectionString =
"Endpoint=sb://azureservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageShar
edAccessKey;SharedAccessKey=<copy-servicebus-sharedkey-from-azure-portal>";
        static string QueuePath = "sbqueue";


        static void Main(string[] args)
        {
            //Service Bus Queue Receiver
            var queueClient = QueueClient.CreateFromConnectionString(ConnectionString,
QueuePath);

            queueClient.OnMessage(msg => ProcessMessage(msg));

            Console.WriteLine("Press Enter to Exit...");
            Console.ReadLine();

            queueClient.Close();
        }

        private static void ProcessMessage(BrokeredMessage msg)
        {
            var text = msg.GetBody<string>();
            Console.WriteLine("\nReceived Messages : " + text);
        }

    }
}
```

Step 14: Now run the Receiver project.


file:///E:/Azure/ServiceBus/SBQueueSender/SBQueueReceiver/bin/Debug/SBQueueReceiver.EXE

```
Press Enter to Exit...
Received Messages : Sender's Message ==> 0
Received Messages : Sender's Message ==> 4
Received Messages : Sender's Message ==> 3
Received Messages : Sender's Message ==> 7
Received Messages : Sender's Message ==> 1
Received Messages : Sender's Message ==> 8
Received Messages : Sender's Message ==> 6
Received Messages : Sender's Message ==> 9
Received Messages : Sender's Message ==> 5
Received Messages : Sender's Message ==> 2
```

Navigate to Azure Portal.



Namespace
azureservicebus

ACTIVE MESSAGE COUNT
0 MESSAGES

SCHEDULED MESSAGE COUNT
0 MESSAGES

DEAD-LETTER MESSAGE COUNT
0 MESSAGES

TRANSFER MESSAGE COUNT
0 MESSAGES

TRANSFER DEAD-LETTER MESSAGE COUNT
0 MESSAGES

MAX SIZE
16 GB

CURRENT
0.0 KB

100% FREE SPACE