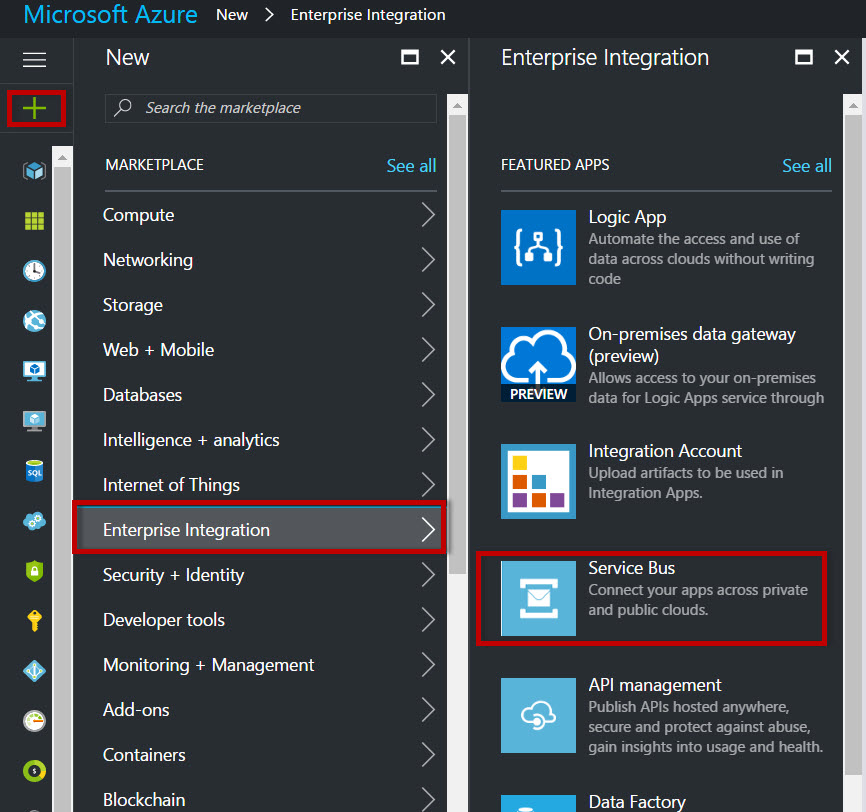
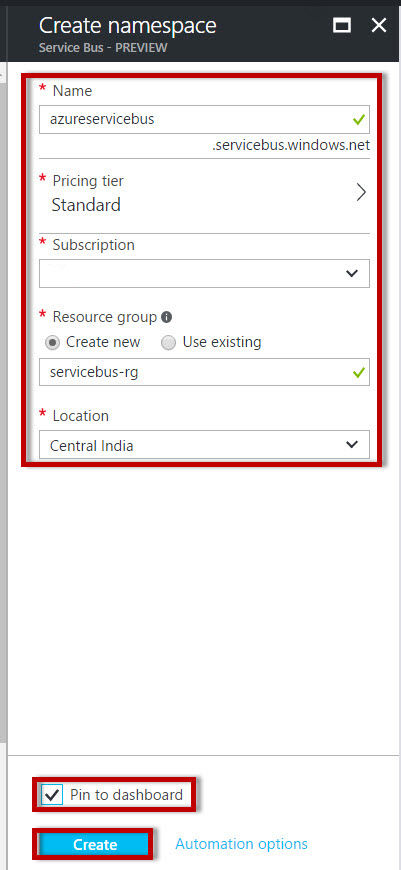
**Azure Service Bus – Relay (Echo Application)**

Step 1: Step 1: Navigate to Azure Portal.

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

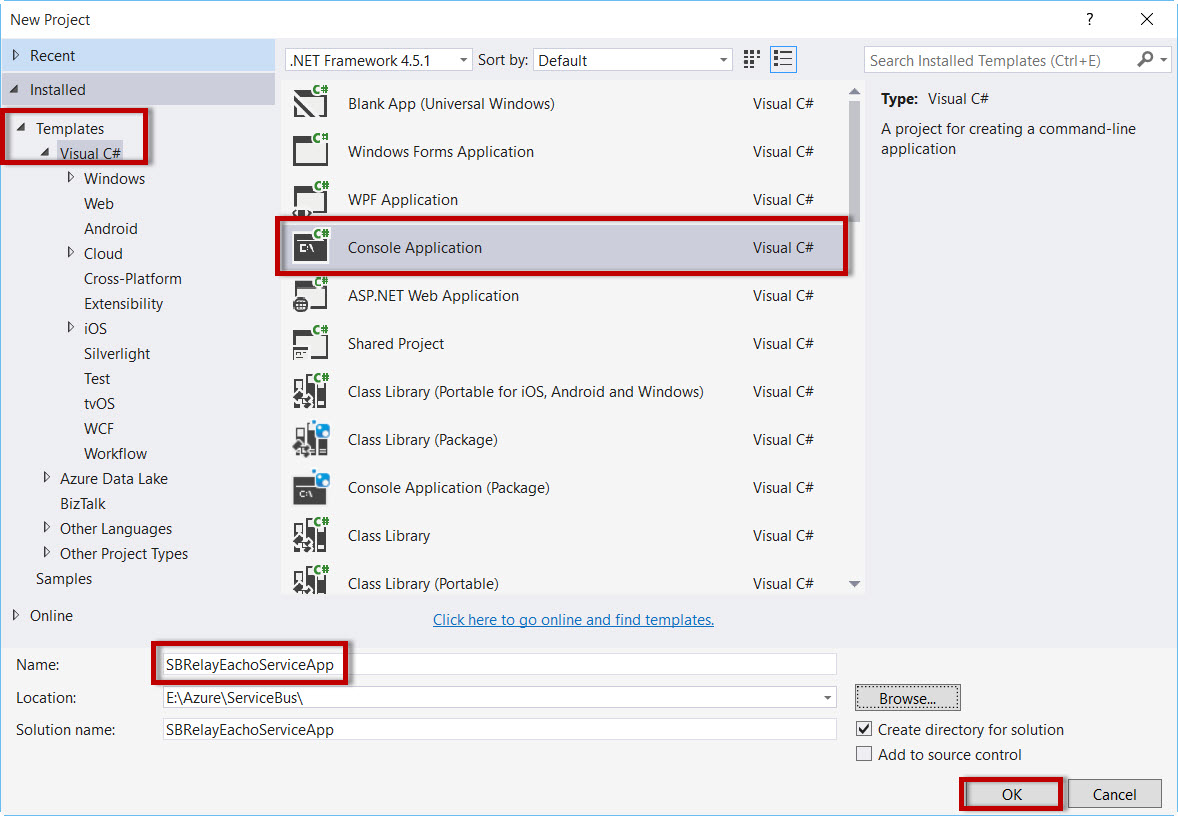


Step 3: Enter Service Bus Namespace Name, Choose Pricing Tier, Resource Group, Location

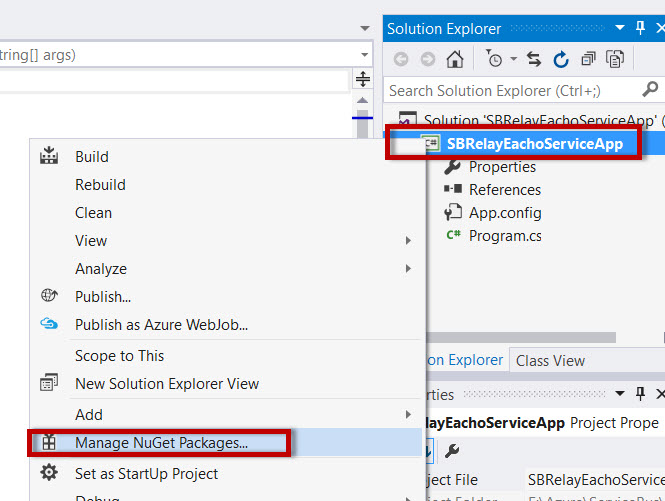


Step 4: Start Visual Studio & Choose Visual C# Template.

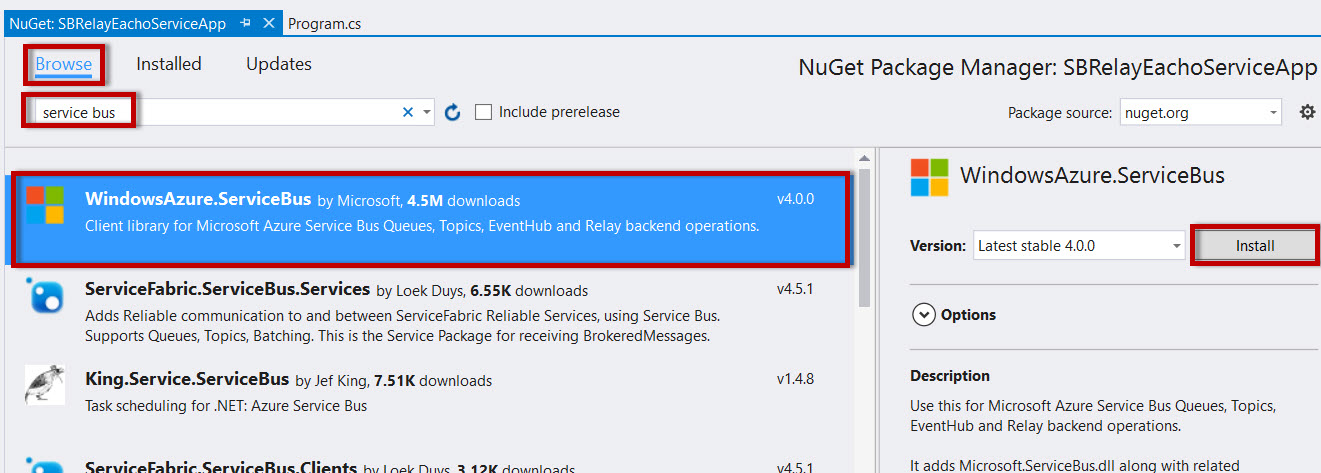
Create New Console Application



Step 5: Right click on Project Name & select “**Manage NuGet Packages…**”



Step 6: Search for “**service bus**” & install **WindowsAzure.ServiceBus** packages



Step 7: Open Program.cs file

Add namespace

using System.ServiceModel;

using System.ServiceModel.Description;

also replace original namespace with new code

namespace SBRelayEachoServiceApp

{

class Program

{

static void Main(string[] args)

{

}

}

}

**to**

namespace Microsoft.ServiceBus.Samples

{

[ServiceContract(Name = "IEchoContract", Namespace = "http://samples.microsoft.com/ServiceModel/Relay/")]

public interface IEchoContract

{

[OperationContract]

String Echo(string text);

}

public interface IEchoChannel : IEchoContract, IClientChannel { };

[ServiceBehavior(Name = "EchoService", Namespace = "http://samples.microsoft.com/ServiceModel/Relay/")]

class EchoService : IEchoContract

{

public string Echo(string text)

{

Console.WriteLine("Echoing: {0}", text);

return text;

}

}

class Program

{

static void Main(string[] args)

{

ServiceBusEnvironment.SystemConnectivity.Mode = ConnectivityMode.AutoDetect;

Console.Write("Your Service Namespace: ");

string serviceNamespace = Console.ReadLine();

Console.Write("Your SAS key: ");

string sasKey = Console.ReadLine();

// Create the credentials object for the endpoint.

TransportClientEndpointBehavior sasCredential = new TransportClientEndpointBehavior();

sasCredential.TokenProvider = TokenProvider.CreateSharedAccessSignatureTokenProvider("RootManageSharedAccessKey", sasKey);

// Create the service URI based on the service namespace.

Uri address = ServiceBusEnvironment.CreateServiceUri("sb", serviceNamespace, "EchoService");

// Create the service host reading the configuration.

ServiceHost host = new ServiceHost(typeof(EchoService), address);

// Create the ServiceRegistrySettings behavior for the endpoint.

IEndpointBehavior serviceRegistrySettings = new ServiceRegistrySettings(DiscoveryType.Public);

// Add the Service Bus credentials to all endpoints specified in configuration.

foreach (ServiceEndpoint endpoint in host.Description.Endpoints)

{

endpoint.Behaviors.Add(serviceRegistrySettings);

endpoint.Behaviors.Add(sasCredential);

}

// Open the service.

host.Open();

Console.WriteLine("Service address: " + address);

Console.WriteLine("Press [Enter] to exit");

Console.ReadLine();

// Close the service.

host.Close();

}

}

}

Step 8: Open App.config file

?xml version="1.0" encoding="utf-8"?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1"/>

</startup>

<system.serviceModel>

<services>

<service name="Microsoft.ServiceBus.Samples.EchoService">

<endpoint contract="Microsoft.ServiceBus.Samples.IEchoContract" binding="netTcpRelayBinding" />

</service>

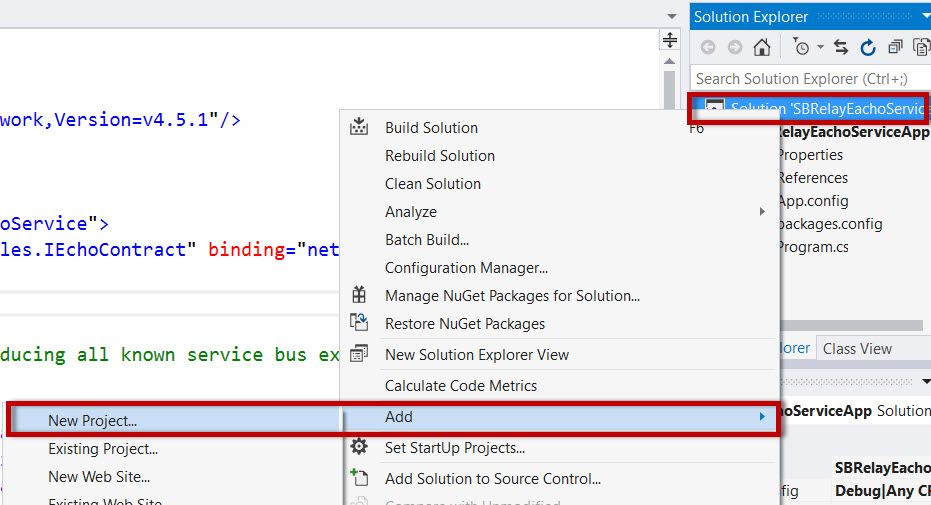
</services>

<extensions>

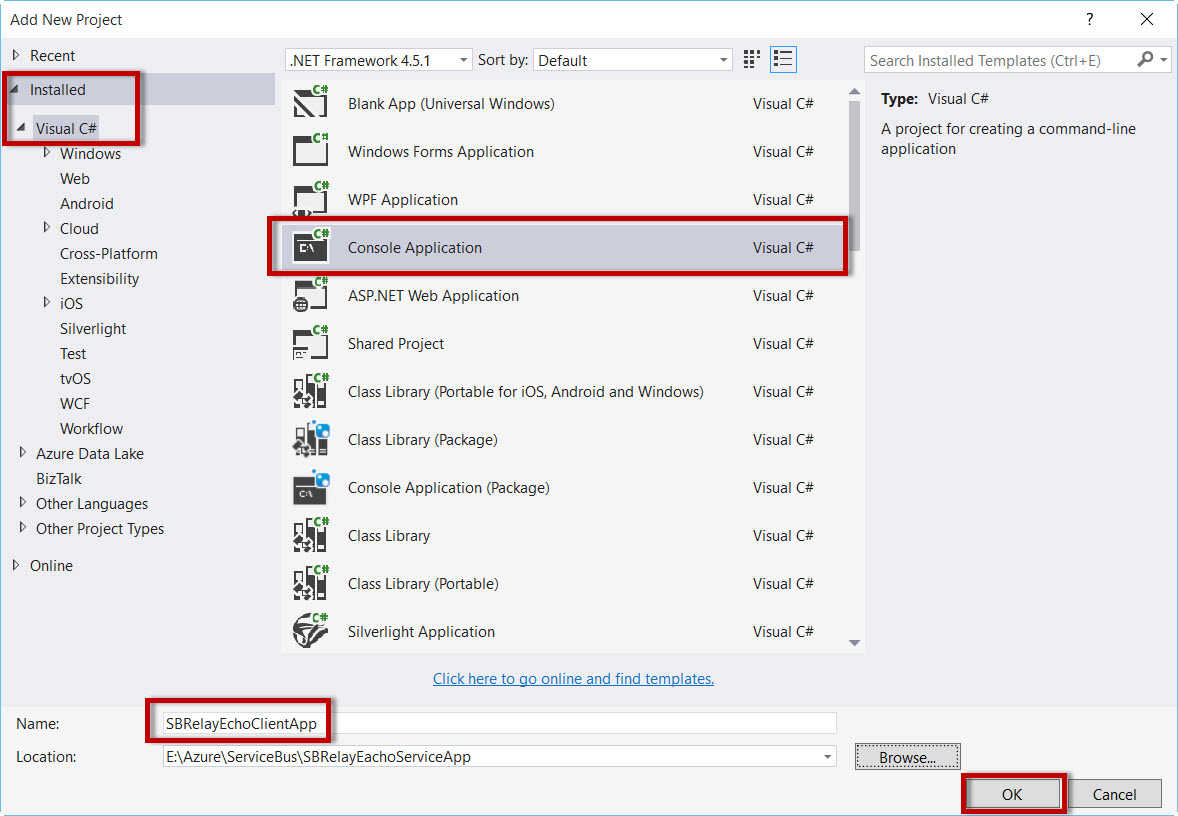
<!-- In this extension section we are introducing all known service bus extensions. User can remove the ones they don't need. -->

<behaviorExtensions>

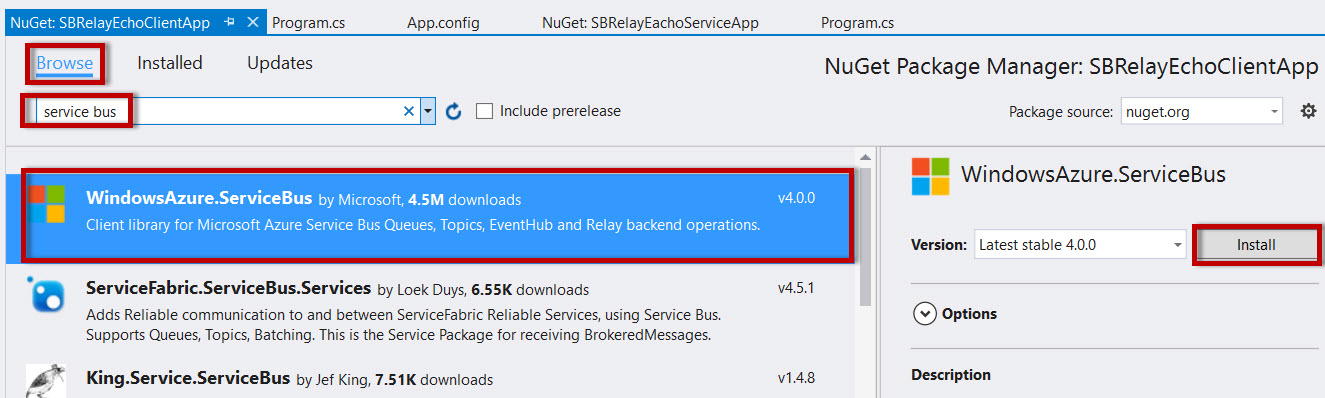
Step 9: Add one more project as a Client App. Right-click on solution name & add new project.



Step 10: Create Client Application



Step 11: Again, install **WindowsAzure.ServiceBus NuGet Package**.



Step 12: Open Program.cs file

Add Namespace

*using System.ServiceModel;*

Remove below code & add yellow highlight code

namespace SBRelayEchoClientApp

{

class Program

{

static void Main(string[] args)

{

}

}

}

**to**

namespace Microsoft.ServiceBus.Samples

{

[ServiceContract(Name = "IEchoContract", Namespace = "http://samples.microsoft.com/ServiceModel/Relay/")]

public interface IEchoContract

{

[OperationContract]

string Echo(string text);

}

public interface IEchoChannel : IEchoContract, IClientChannel { }

class Program

{

static void Main(string[] args)

{

ServiceBusEnvironment.SystemConnectivity.Mode = ConnectivityMode.AutoDetect;

Console.Write("Your Service Namespace: ");

string serviceNamespace = Console.ReadLine();

Console.Write("Your SAS Key: ");

string sasKey = Console.ReadLine();

Uri serviceUri = ServiceBusEnvironment.CreateServiceUri("sb", serviceNamespace, "EchoService");

TransportClientEndpointBehavior sasCredential = new TransportClientEndpointBehavior();

sasCredential.TokenProvider = TokenProvider.CreateSharedAccessSignatureTokenProvider("RootManageSharedAccessKey", sasKey);

ChannelFactory<IEchoChannel> channelFactory = new ChannelFactory<IEchoChannel>("RelayEndpoint", new EndpointAddress(serviceUri));

channelFactory.Endpoint.Behaviors.Add(sasCredential);

IEchoChannel channel = channelFactory.CreateChannel();

channel.Open();

Console.WriteLine("Enter text to echo (or [Enter] to exit):");

string input = Console.ReadLine();

while (input != String.Empty)

{

try

{

Console.WriteLine("Server echoed: {0}", channel.Echo(input));

}

catch (Exception e)

{

Console.WriteLine("Error: " + e.Message);

}

input = Console.ReadLine();

}

channel.Close();

channelFactory.Close();

}

}

}

Step 13: Open App.config file

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1"/>

</startup>

<system.serviceModel>

<client>

<endpoint name="RelayEndpoint"

contract="Microsoft.ServiceBus.Samples.IEchoContract"

binding="netTcpRelayBinding"/>

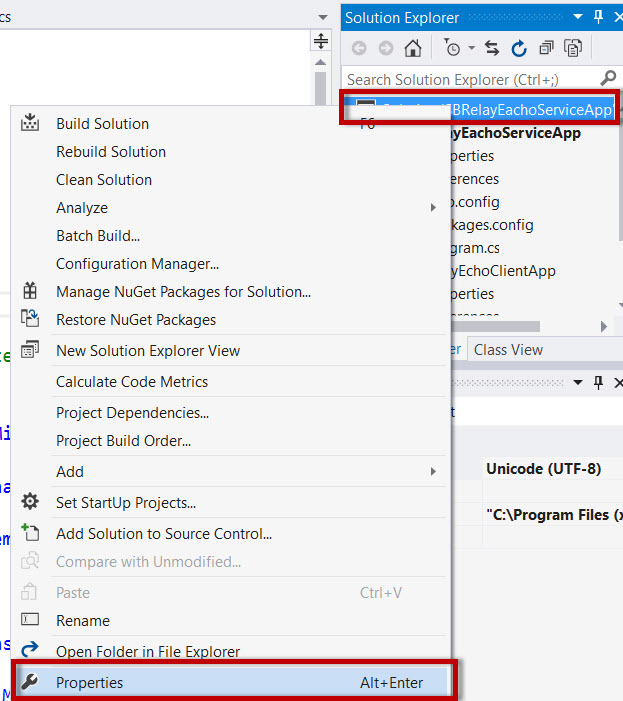
</client>

<extensions>

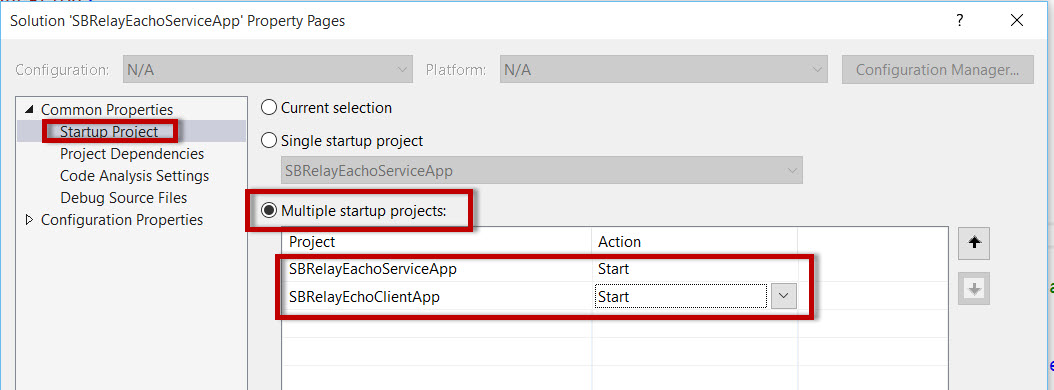
<!-- In this extension section we are introducing all known service bus extensions. User can remove the ones they don't need. -->

<behaviorExtensions>

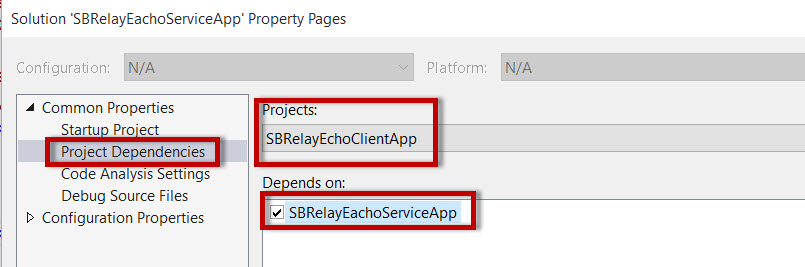
Step 14: Right-click on solution name & select Properties.



Select Multiple startup projects into “Startup Project” option



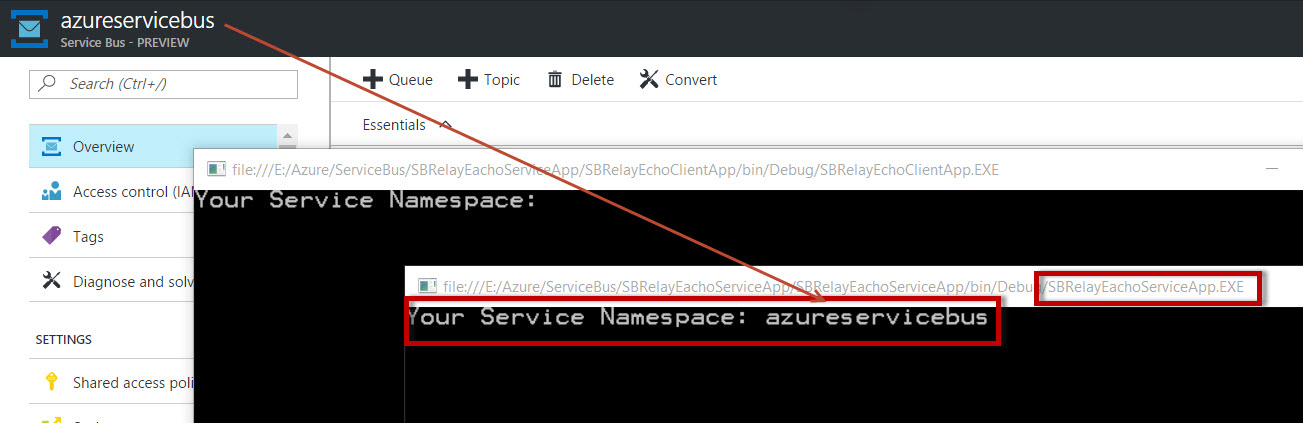
Also change Project Dependencies

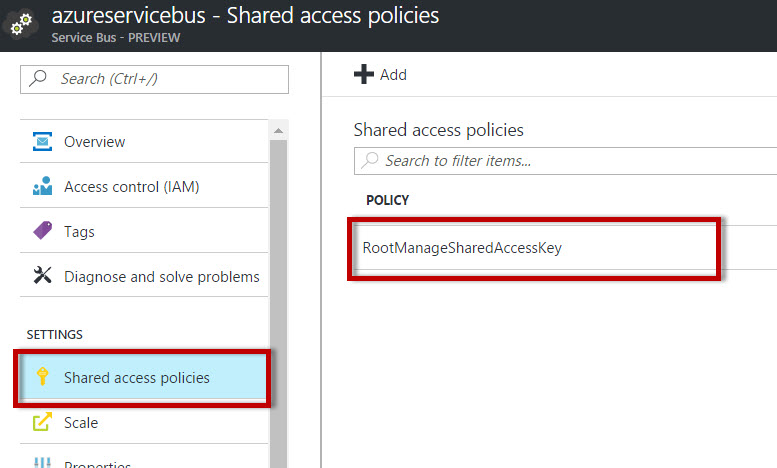


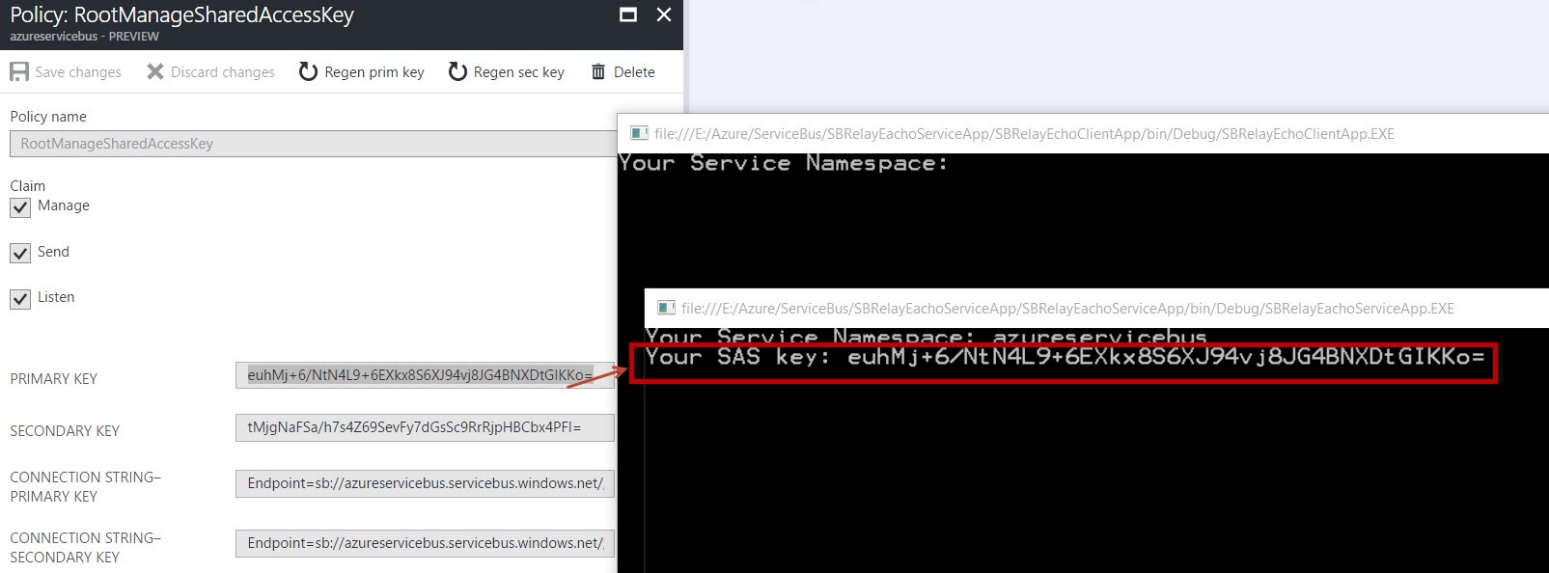
Step 15: Run the project

Two Console Application will run(Service & Client)

First select Service Application & enter service bus namespace & Key from Portal.







Navigate to Client Application. Enter namespace & Key from Portal.

Enter some text

