ffhtt Ctt tffhC ttjC e tt

Friday, January 30, 2015

L ttr fb ttr

FabAct is a virtual actor based programming model for Windows Fabric. Windows Fabric is a platform for building highly reliable, scalable applications for both cloud and on premise that are easy to develop and manage. Windows Fabric is used by many Microsoft first party applications on both on premises and cloud; such as, Lync, Cortana, MSODS, SQL Azure, SQL BI, Document DB, Azure Resource Providers, InTune.

Cttj ffCh GCC ffh

There are 3 samples demonstrating the Actor based programming model.

- HelloWorld
- Calculator
- Voicemailbox

In each Sample directory there is a PowerShell script (ManageActorApplication.ps1) that can be used to deploy the application on the local cluster.

e ttC r Ctt C ffhCtt tffhC ttjCC ffhC r ffhtt

Note: Windows Fabric Actor Model CTP-1 includes an early build of the Visual Studio tools for Windows Fabric. There is no need to use the FabActUtil if you are using the Visual Studio Tools Preview. For Visual Studio 2015 you can use the following instructions to create actor services project.

1) Create a new Visual Studio Solution.

2) Create a new Class Library Project for Actor Interfaces (ensure that it is targeted for x64 platform).

   a. Add project reference to “System.Fabric.Services” from FabricActSDK\bin directory.
   b. Add “using System.Fabric.Services.Actor” to the Interface .cs file.
   c. Inherit the actor interface from IActor: “public interface IMyActor : IActor.”

3) Generate the Service Project: run FabActUtil.exe /t:Project /ap:<application name> /sp:<service package name> in the solution folder. This will generate the Service project and empty manifest files.

4) Add existing project: import Service Project into the solution.

5) Set the build dependency: Service project depends on the Actor Interface project.

6) Add Project References to the Service project and add Interface project reference.

7) Add a default namespace to the Service Project.

8) Open the build Configuration Manager and ensure that every project is building for x64.

9) Implement your Actor by adding a new Class to the Service Project.

   a. Add “using System.Fabric.Services.Actor” to the Interface .cs file.
   b. Inherit the actor interface from IActor: “public interface IMyActor : IActor.”
   c. Inherit the actor class from the Actor and interface IMyActor: “public class MyActor : Actor,

IMyActor."

d. Register the actor implementation with FabricRuntime in Program.cs

10) Rebuild the solution – you should only get errors with respect to not implementing the actor interface methods in the service project. Implement those methods and recompile.

11) To deploy the application to the location cluster, use -Deploy command from PowerShell script ManageActorApp.ps1. (ManageActorApp.ps1 -Deploy)

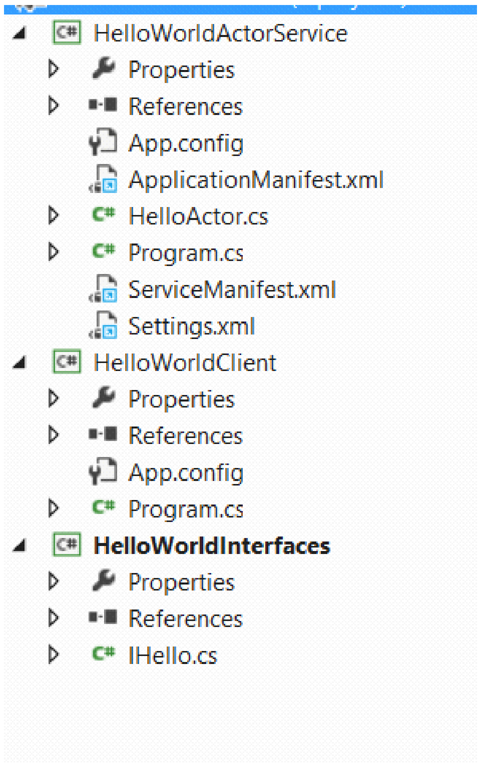
12) Create a Console Client project by adding a new Project: Console Application.

Vj ffCh ffhr A r flC

Thursday, January 15, 2015 2:19 PM

Vj ffCh ffhr A r flC ffh

The classic HelloWorld sample that shows the basic structure of a FabAct application. You should start with this sample to understand the basics of how structure the code to define Actors, Actor Interfaces and the client application.



```
◢ HelloWorldActorService
    ▷ Properties
    ▷ References
      App.config
      ApplicationManifest.xml
    ▷ HelloActor.cs
    ▷ Program.cs
      ServiceManifest.xml
      Settings.xml
◢ HelloWorldClient
    ▷ Properties
    ▷ References
      App.config
    ▷ Program.cs
◢ HelloWorldInterfaces
    ▷ Properties
    ▷ References
    ▷ IHello.cs
```

R ffh ffhCGffh ttr

The HelloSample shows the typical structure of a FabAct-based application. The solution is composed of 3 projects:

- **The Service (HelloWorldActorService):** this is the code that is used to 'expose' the application as a Windows Fabric service. It include the implementation of the Actor interface and some basic initialization in the program.cs file. These are usually the two components that needs to be modified.

- **The Actor Interfaces (HelloWorldInterfaces):** the project contains the definition of the interface(s) that that the actor exposes. In the HelloWorld sample the Actor exposes only one method (SayHello) that is implemented in the HelloWorldActorService project.

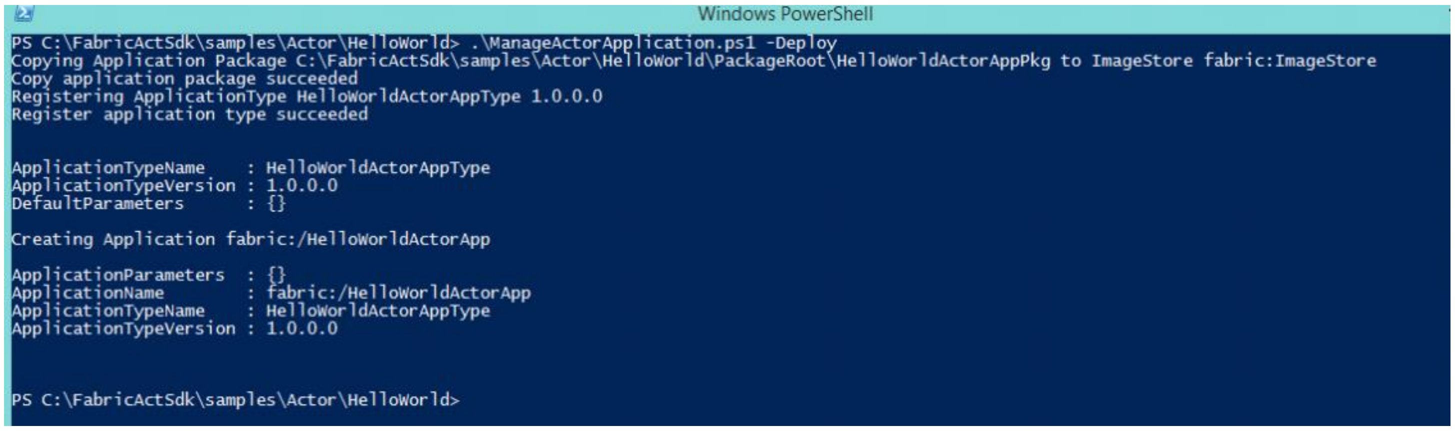
- **The Client application (HelloWorldClient)**: this is the application that uses the ActorProxy class to instantiate and invoke methods on the newly created Actor. In the code example below a random ID is generated which corresponds to an actor in the -int64 to +int64 number range. The SayHello method is called on the actor and concatenated string printed out to the console.

```
var friend = ActorProxy.Create<IHello>(ActorId.GetRandom(), ApplicationName);
Console.WriteLine("\n\nFrom Actor {1}: {0}\n\n", friend.SayHello("Good morning!").Result, friend.GetActorId());
```

Ctj ffCh ffh

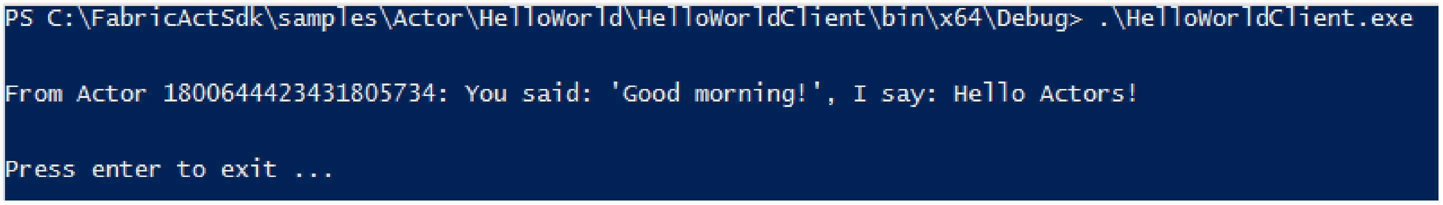
First step is to build the solution. The build script will compile the code and run the FabActUtil to package the code and manifest files so it can be deployed on the Windows Fabric dev cluster.

1. **Deploy the HelloWorld actor application on to the local cluster.**
    a. Open a new powershell window (you do not need to open it as administrator).
    b. Go to the Samples\Actor\HelloWorld folder and run `.\ManageActorApplication.ps1 -Deploy`



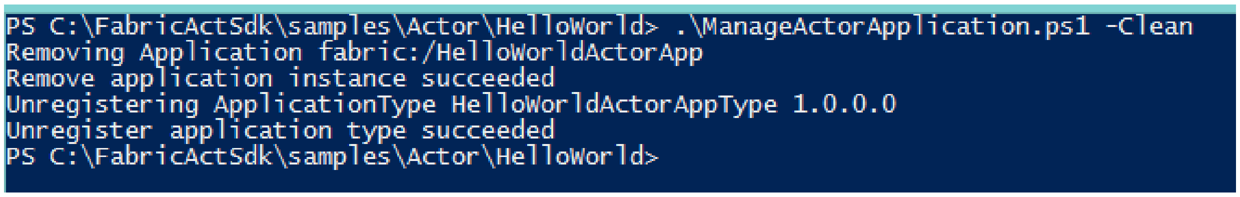
```
PS C:\FabricActSdk\samples\Actor\HelloWorld> .\ManageActorApplication.ps1 -Deploy
Copying Application Package C:\FabricActSdk\samples\Actor\HelloWorld\PackageRoot\HelloWorldActorAppPkg to ImageStore fabric:ImageStore
Copy application package succeeded
Registering ApplicationType HelloWorldActorAppType 1.0.0.0
Register application type succeeded


ApplicationTypeName    : HelloWorldActorAppType
ApplicationTypeVersion : 1.0.0.0
DefaultParameters      : {}

Creating Application fabric:/HelloWorldActorApp

ApplicationParameters  : {}
ApplicationName        : fabric:/HelloWorldActorApp
ApplicationTypeName    : HelloWorldActorAppType
ApplicationTypeVersion : 1.0.0.0


PS C:\FabricActSdk\samples\Actor\HelloWorld>
```

2. **Start the HelloWorld Actor Client.**
    a. From the powershell go to HelloWorldClient\bin\x64\Debug and launch HelloWorldClient.exe



```
PS C:\FabricActSdk\samples\Actor\HelloWorld\HelloWorldClient\bin\x64\Debug> .\HelloWorldClient.exe

From Actor 1800644423431805734: You said: 'Good morning!', I say: Hello Actors!

Press enter to exit ...
```

3. **Remove the HelloWorld Actor application**
    a. From the powershell windows run the following command to remove and unregister the hello world application from the local cluster
    b. `.\ManageActorApplication.ps1 -Clean`



```
PS C:\FabricActSdk\samples\Actor\HelloWorld> .\ManageActorApplication.ps1 -Clean
Removing Application fabric:/HelloWorldActorApp
Remove application instance succeeded
Unregistering ApplicationType HelloWorldActorAppType 1.0.0.0
Unregister application type succeeded
PS C:\FabricActSdk\samples\Actor\HelloWorld>
```

- Project Structure: Service, Actor, ActorInterface classes
- Understand the build process: application code vs. generated code
- Change code, build, deploy, test/debug
- Calculator: another example of stateless actors
    - Show both the client app and REST-APIs scenarios (REST using project K)

Vj ffCh tt C ffh

This sample shows how a stateless actor is defined, implemented and used. Like the HelloWorld sample, the solution folder has three projects:

`CalculatorInterfaces` defines the stateless calculator actor.

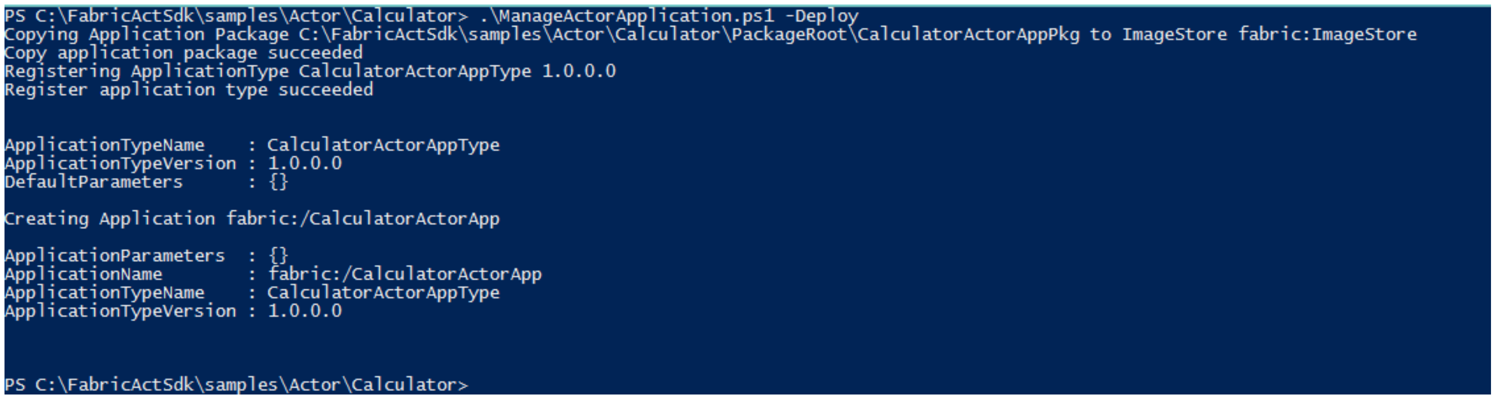
`CalculatorActorService` provides the implementation of the actor. It also includes generated code including the bases classes for implementing the Calculator actor.

`CalculatorClient` shows how to send messages to the calculator actor. The code generates a random actor id (showing that the actors are virtual) and asks it to add two numbers. It does this in the loop continuously. The number printed on the screen when you run the client is the number of operations performed so far.

Ctj ffCh ffh

Using the same process used for the HelloWorld, you can deploy the Calculator Sample by running the ManageActorApplication.ps1 script under the samples\Actor\Calculator directory.

run `.\ManageActorApplication.ps1 -Deploy`

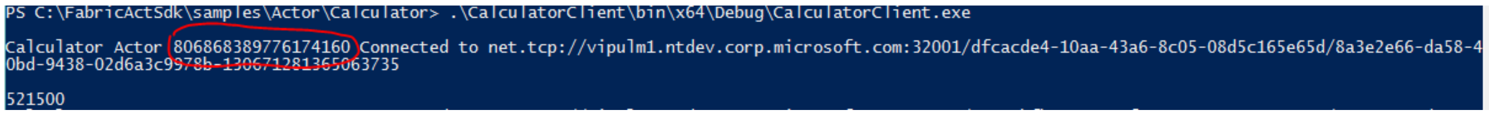


```
PS C:\FabricActSdk\samples\Actor\Calculator> .\ManageActorApplication.ps1 -Deploy
Copying Application Package C:\FabricActSdk\samples\Actor\Calculator\PackageRoot\CalculatorActorAppPkg to ImageStore fabric:ImageStore
Copy application package succeeded
Registering ApplicationType CalculatorActorAppType 1.0.0.0
Register application type succeeded


ApplicationTypeName    : CalculatorActorAppType
ApplicationTypeVersion : 1.0.0.0
DefaultParameters      : {}

Creating Application fabric:/CalculatorActorApp

ApplicationParameters  : {}
ApplicationName        : fabric:/CalculatorActorApp
ApplicationTypeName    : CalculatorActorAppType
ApplicationTypeVersion : 1.0.0.0



PS C:\FabricActSdk\samples\Actor\Calculator>
```

1. **Start the Calculator Actor Client.**

   From the powershell window (assuming you are using the one opened above), do "`start .\CalculatorClient\bin\x64\Debug\CalculatorClient.exe`"

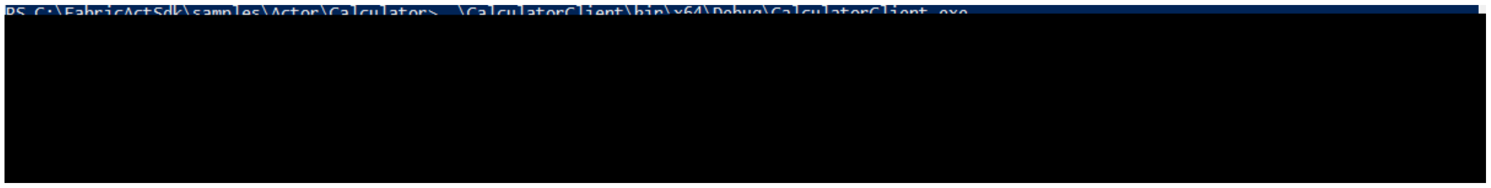
   This should bring up a command window running the CalculatorClient.



```
PS C:\FabricActSdk\samples\Actor\Calculator> .\CalculatorClient\bin\x64\Debug\CalculatorClient.exe

Calculator Actor 806868389776174160 Connected to net.tcp://vipulm1.ntdev.corp.microsoft.com:32001/dfcacde4-10aa-43a6-8c05-08d5c165e65d/8a3e2e66-da58-40bd-9438-02d6a3c9978b-130671281365063735

521500
```

   Introduce the failover by restarting the replica that hosts this particular calculator actor instance. Use the actor id from the place circled red in the Window above. Note that actor ids are generated randomly, so actor id on your machine would be different and can be a negative number. While the CalculatorClient app is still running, on a new powershell windows run the following script:

You should see the Calculator Client stop for a moment, then reconnect to the failed over actor and continue. This is indicated be a different tcp endpoint address as shown in the screen shot below.



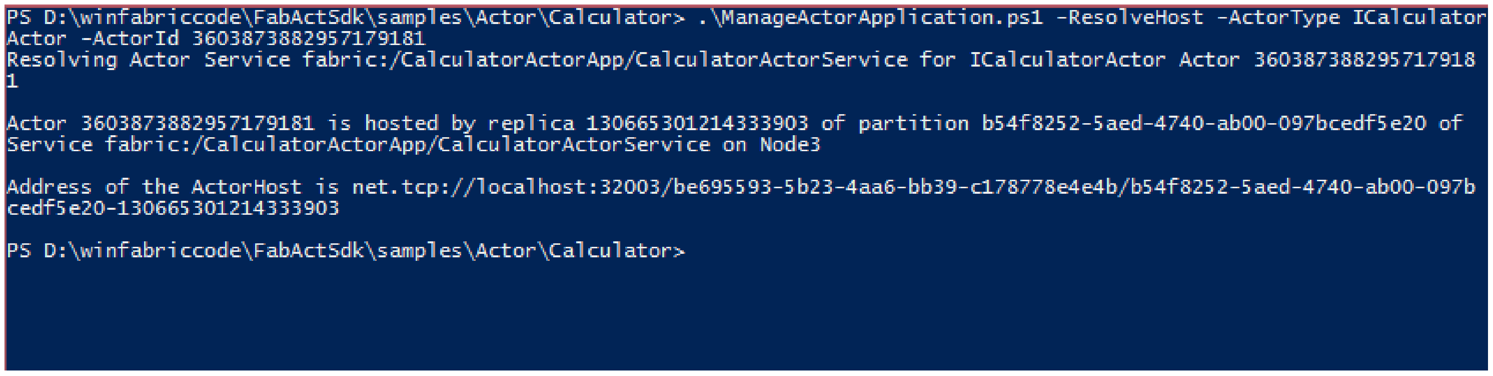
```
PS C:\FabricActSdk\samples\Actor\Calculator> .\CalculatorClient\bin\x64\Debug\CalculatorClient.exe
```

[redacted]

The second 'Calculator Actor …' line shows that the Actor was failed over seamlessly to failed over process.

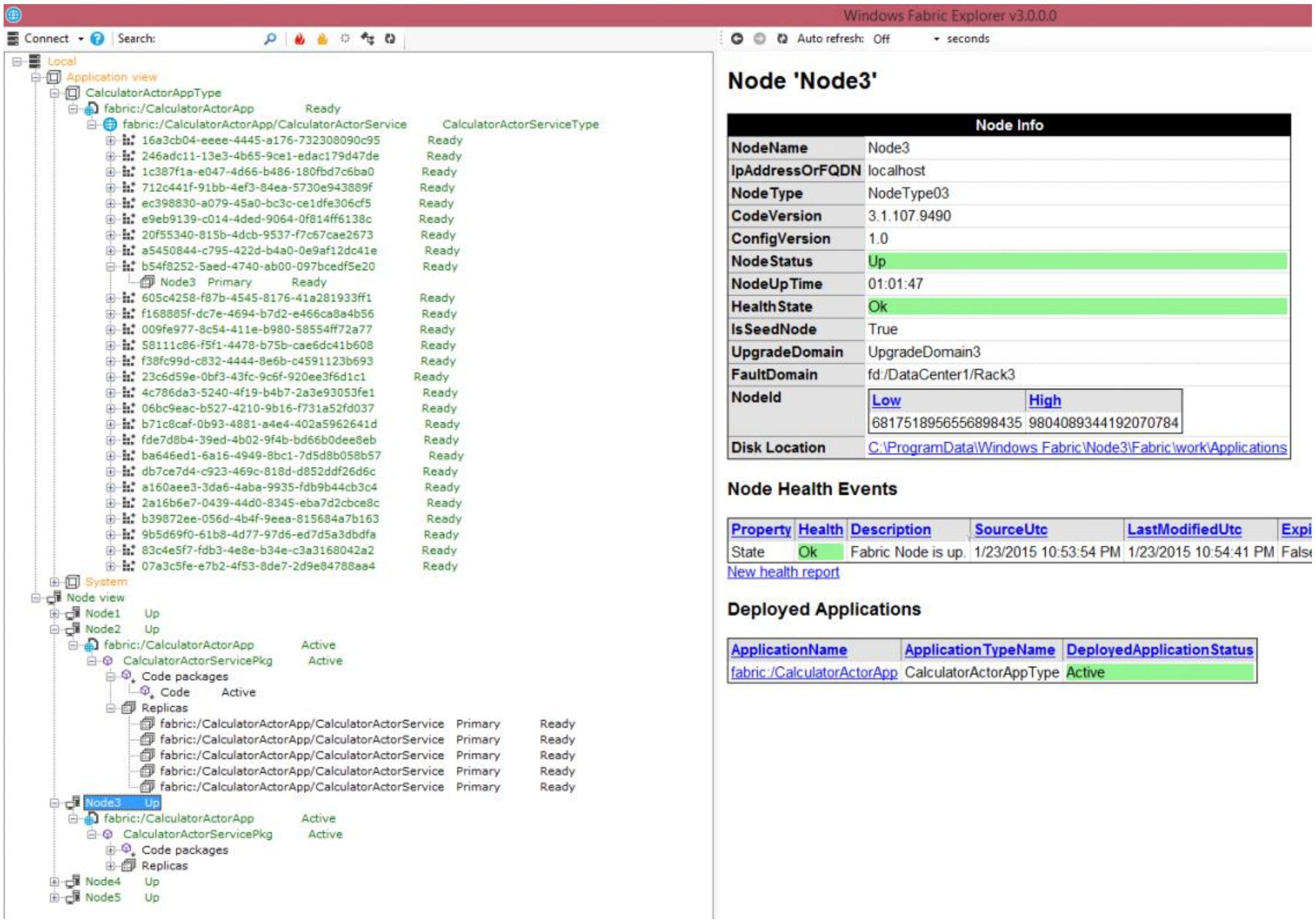
You can also introduce the failover by killing the CalculatorActorService.exe on the right node. To see the Node name on which the actor is hosted, you can run -ResolveHost command.

```
.\ManageActorApplication.ps1 -ResolveHost –ActorType ICalculatorActor -
ActorId  <actorId>
```



```
PS D:\winfabriccode\FabActSdk\samples\Actor\Calculator> .\ManageActorApplication.ps1 -ResolveHost -ActorType ICalculator
Actor -ActorId 3603873882957179181
Resolving Actor Service fabric:/CalculatorActorApp/CalculatorActorService for ICalculatorActor Actor 360387388295717918
1

Actor 3603873882957179181 is hosted by replica 130665301214333903 of partition b54f8252-5aed-4740-ab00-097bcedf5e20 of
Service fabric:/CalculatorActorApp/CalculatorActorService on Node3

Address of the ActorHost is net.tcp://localhost:32003/be695593-5b23-4aa6-bb39-c178778e4e4b/b54f8252-5aed-4740-ab00-097b
cedf5e20-130665301214333903

PS D:\winfabriccode\FabActSdk\samples\Actor\Calculator>
```

Now launch winfabexplorer.exe tool and connect to the local cluster. You should see a screen similar to below. Here you can see the partition (b54fb252…) that replica for the actor Id (360387…) is in and that this replica has been placed on Node3 in the cluster.



```
Local
  Application view
    CalculatorActorAppType
      fabric:/CalculatorActorApp          Ready
        fabric:/CalculatorActorApp/CalculatorActorService          CalculatorActorServiceType
          16a3cb04-eeee-4445-a176-732308090c95     Ready
          246adc11-13e3-4b65-9ce1-edac179d47de     Ready
          1c387f1a-e047-4d66-b486-180fbd7c6ba0     Ready
          712c441f-91bb-4ef3-84ea-5730e943889f     Ready
          ec398830-a079-45a0-bc3c-ce1dfe306cf5     Ready
          e9eb9139-c014-4ded-9064-0f814ff6138c     Ready
          20f55340-815b-4dcb-9537-f7c67cae2673     Ready
          a5450844-c795-422d-b4a0-0e9af12dc41e     Ready
          b54f8252-5aed-4740-ab00-097bcedf5e20     Ready
            Node3   Primary        Ready
          605c4258-f87b-4545-8176-41a281933ff1     Ready
          f168885f-dc7e-4694-b7d2-e466ca8a4b56     Ready
          009fe977-8c54-411e-b980-58554ff72a77     Ready
          58111c86-f5f1-4478-b75b-cae6dc41b608     Ready
          f38fc99d-c832-4444-8e6b-c4591123b693     Ready
          23c6d59e-0bf3-43fc-9c6f-920ee3f6d1c1     Ready
          4c786da3-5240-4f19-b4b7-2a3e93053fe1     Ready
          06bc9eac-b527-4210-9b16-f731a52fd037     Ready
          b71c8caf-0b93-4881-a4e4-402a5962641d     Ready
          fde7d8b4-39ed-4b02-9f4b-bd66b0dee8eb     Ready
          ba646ed1-6a16-4949-8bc1-7d5d8b058b57     Ready
          db7ce7d4-c923-469c-818d-d852ddf26d6c     Ready
          a160aee3-3da6-4aba-9935-fdb9b44cb3c4     Ready
          2a16b6e7-0439-44d0-8345-eba7d2cbce8c     Ready
          b39872ee-056d-4b4f-9eea-815684a7b163     Ready
          9b5d69f0-61b8-4d77-97d6-ed7d5a3dbdfa     Ready
          83c4e5f7-fdb3-4e8e-b34e-c3a3168042a2     Ready
          07a3c5fe-e7b2-4f53-8de7-2d9e84788aa4     Ready
    System
  Node view
    Node1     Up
    Node2     Up
      fabric:/CalculatorActorApp          Active
        CalculatorActorServicePkg          Active
          Code packages
            Code        Active
          Replicas
            fabric:/CalculatorActorApp/CalculatorActorService   Primary        Ready
            fabric:/CalculatorActorApp/CalculatorActorService   Primary        Ready
            fabric:/CalculatorActorApp/CalculatorActorService   Primary        Ready
            fabric:/CalculatorActorApp/CalculatorActorService   Primary        Ready
            fabric:/CalculatorActorApp/CalculatorActorService   Primary        Ready
    Node3     Up
      fabric:/CalculatorActorApp          Active
        CalculatorActorServicePkg          Active
          Code packages
          Replicas
    Node4     Up
    Node5     Up
```

Node 'Node3'

Node Info	
NodeName	Node3
IpAddressOrFQDN	localhost
NodeType	NodeType03
CodeVersion	3.1.107.9490
ConfigVersion	1.0
NodeStatus	Up
NodeUpTime	01:01:47
HealthState	Ok
IsSeedNode	True
UpgradeDomain	UpgradeDomain3
FaultDomain	fd:/DataCenter1/Rack3
NodeId	Low: 6817518956556898435 / High: 9804089344192070784
Disk Location	C:\ProgramData\Windows Fabric\Node3\Fabric\work\Applications

Node Health Events

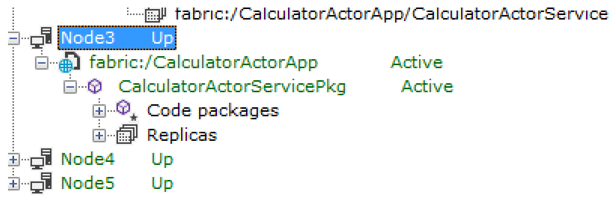
Property	Health	Description	SourceUtc	LastModifiedUtc	Expi
State	Ok	Fabric Node is up.	1/23/2015 10:53:54 PM	1/23/2015 10:54:41 PM	False

New health report

Deployed Applications

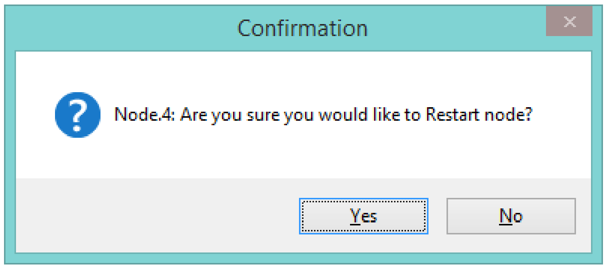
ApplicationName	ApplicationTypeName	DeployedApplicationStatus
fabric:/CalculatorActorApp	CalculatorActorAppType	Active

Navigate to Node3 in the WinfabExplorer tool and right click on the node. Select the "Start / Stop - NOT SUPPORTED FOR PRODUCTION USE" and then select Restart Node.



```
          fabric:/CalculatorActorApp/CalculatorActorService
Node3     Up
  fabric:/CalculatorActorApp          Active
    CalculatorActorServicePkg          Active
      Code packages
      Replicas
Node4     Up
Node5     Up
```

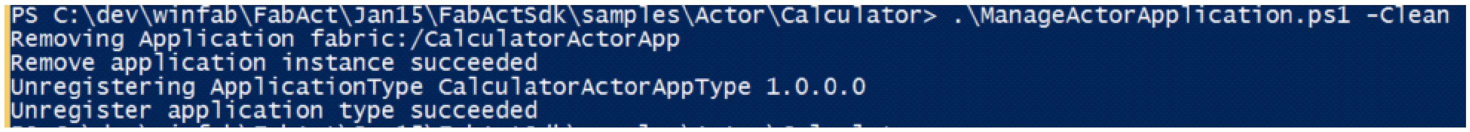
A confirmation dialog box pops up. Before clicking yes, go open the PowerShell window that is running the CalculatorClient. Then click Yes and you will see the same failover happen again for the Calculator client. This has done the same failover but through the tool rather than a PowerShell command.



Node.4: Are you sure you would like to Restart node?

3. **Remove the calculator app from the cluster.**
   In order to reset the demonstration, you should clean the deployment of the Calculator Actor from the cluster by calling the following command.

```
.\ManageActorApplication.ps1 -Clean
```



```
PS C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\Calculator> .\ManageActorApplication.ps1 -Clean
Removing Application fabric:/CalculatorActorApp
Remove application instance succeeded
Unregistering ApplicationType CalculatorActorAppType 1.0.0.0
Unregister application type succeeded
```

This is also required if you want to make changes to the code and re-deploy the modified code to the cluster. Note that in production you will perform an application upgrade, but here we are removing the application from the cluster and redeploying it.

ﬀbﬂC tt tﬀﬁtt C tt

Thursday, January 15, 2015 2:20 PM

tt tﬀh C r ﬀh r ᶜC tt

This sample shows how to define, implement and use an actor that has persistent state. The actor is a VoicemailBox of a custo mer represented by Actor Id. It store the voicemails for that customer, it allows leaving a new message, retrieving the messages, setting the new greeting, deleting a particular message and deleting all messages. The solution folder has three projects:

`VoicemailBoxInterfaces` defines the stateful actor and its state.

`VoicemailBoxActorService` provides the implementation of the actor. It also includes generated code including the bases classes for implementing the VoicemailBox actor.

`VoicemailBoxActorServiceV2` provides the modified implementation of the actor that has a different default greeting. This is used to demonstrate rolling upgrade capabilities of Windows Fabric. The script is using the unmonitored upgrade but the monitored upgrade with health che ck is also fully supported.

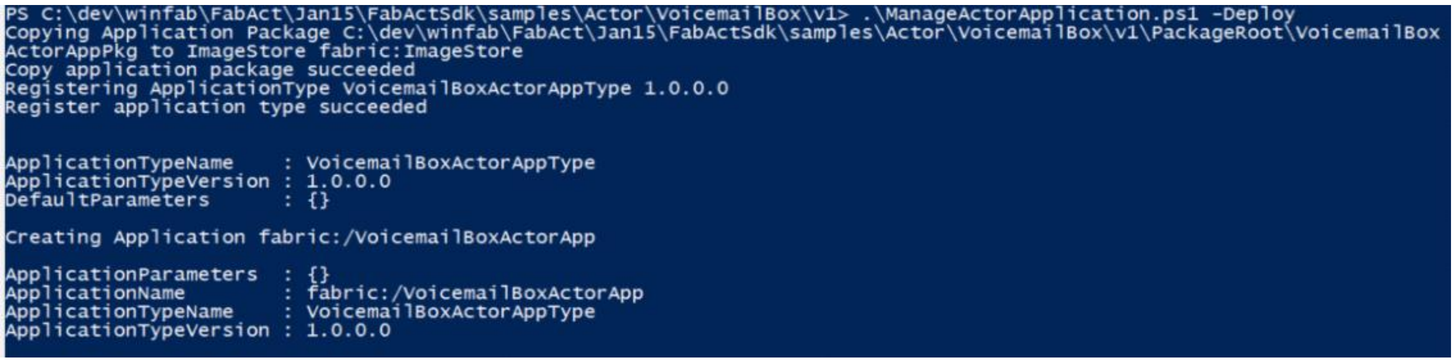
`VoicemailBoxClient` shows how to send messages to the VoicemailBox actor. The code generates a random actor id (showing that the actors are virtu al) and performs various operations on that voicemail box. The code gets the greeting, displays which actor host it is connected to, leaves a new message, displays the current messages in the voicemailbox, and after a certain random number of messages accumulate it either performs a delete or delete all. It also includes generated code including the bases classes for implementing the VoicemailBox actor.

Ctj ﬀC ﬀh

The demonstration of the VoicemailBox actor sample is very similar to the CalculatorActor. The complete instructions are incl uded below.

1. **Deploy the VoicemailBoxV1 actor on to the local cluster.**

   You need to use the ManageActorApplication script under the samples\Actor\VoicemailBox\V1

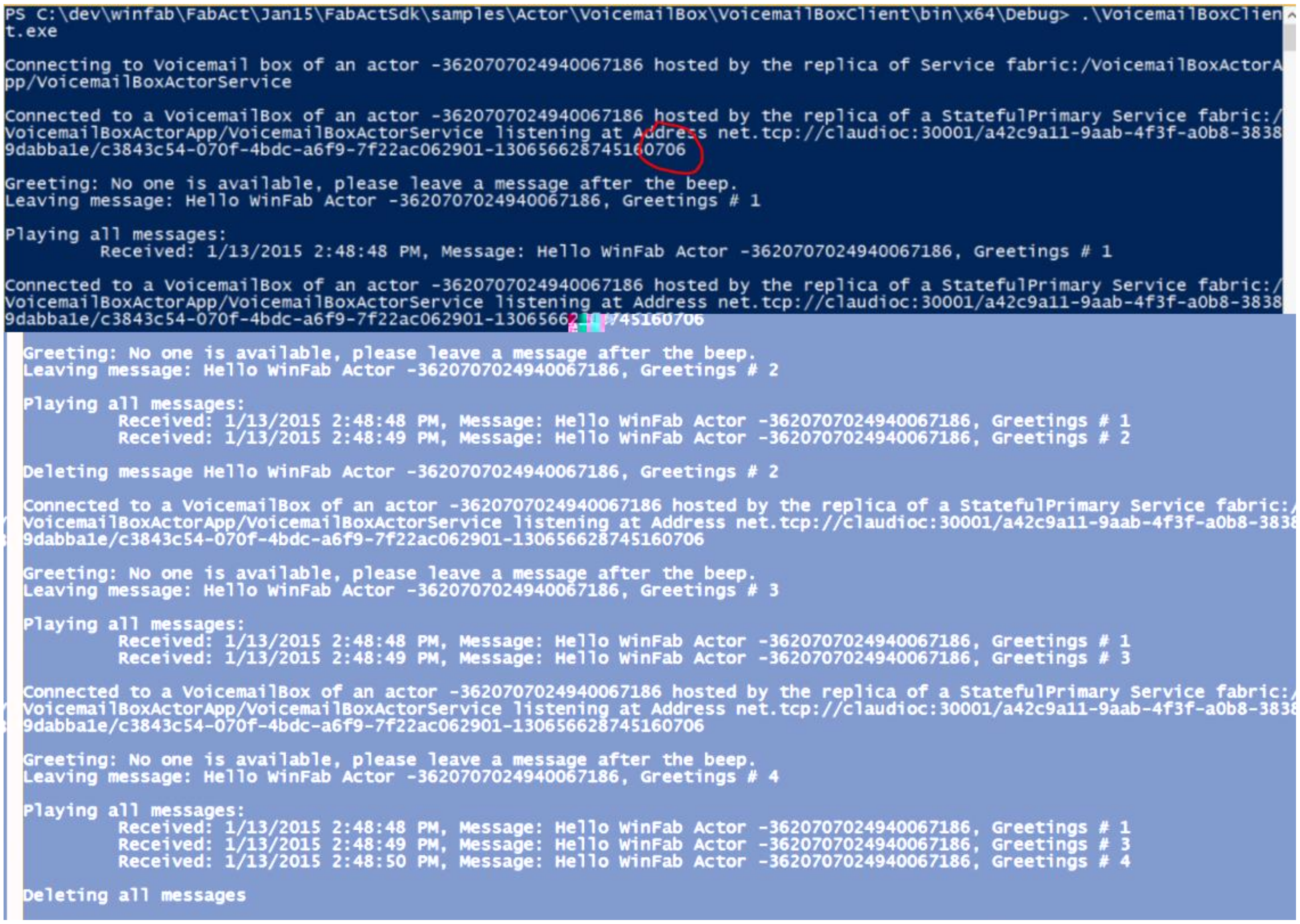


```
PS C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\v1> .\ManageActorApplication.ps1 -Deploy
Copying Application Package C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\v1\PackageRoot\VoicemailBox
ActorAppPkg to ImageStore fabric:ImageStore
Copy application package succeeded
Registering ApplicationType VoicemailBoxActorAppType 1.0.0.0
Register application type succeeded


ApplicationTypeName    : VoicemailBoxActorAppType
ApplicationTypeVersion : 1.0.0.0
DefaultParameters      : {}

Creating Application fabric:/VoicemailBoxActorApp

ApplicationParameters  : {}
ApplicationName        : fabric:/VoicemailBoxActorApp
ApplicationTypeName    : VoicemailBoxActorAppType
ApplicationTypeVersion : 1.0.0.0
```

2. **Start the VoicemailBox Actor Client.**

   From the powershell window (assuming you are using the one opened above), do
   "`start .\VoicemailBoxActorClient\bin\x64\Debug\VoicemailBoxActorClient.exe`"

   This should bring up a command window running the VoicemailBoxActorClient.



```
PS C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\VoicemailBoxClient\bin\x64\Debug> .\VoicemailBoxClien
t.exe

Connecting to Voicemail box of an actor -3620707024940067186 hosted by the replica of Service fabric:/VoicemailBoxActorA
pp/VoicemailBoxActorService

Connected to a VoicemailBox of an actor -3620707024940067186 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:30001/a42c9a11-9aab-4f3f-a0b8-3838
9dabba1e/c3843c54-070f-4bdc-a6f9-7f22ac062901-130656628745160706

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3620707024940067186, Greetings # 1

Playing all messages:
        Received: 1/13/2015 2:48:48 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 1

Connected to a VoicemailBox of an actor -3620707024940067186 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:30001/a42c9a11-9aab-4f3f-a0b8-3838
9dabba1e/c3843c54-070f-4bdc-a6f9-7f22ac062901-130656628745160706

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3620707024940067186, Greetings # 2

Playing all messages:
        Received: 1/13/2015 2:48:48 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 1
        Received: 1/13/2015 2:48:49 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 2

Deleting message Hello WinFab Actor -3620707024940067186, Greetings # 2

Connected to a VoicemailBox of an actor -3620707024940067186 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:30001/a42c9a11-9aab-4f3f-a0b8-3838
9dabba1e/c3843c54-070f-4bdc-a6f9-7f22ac062901-130656628745160706

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3620707024940067186, Greetings # 3

Playing all messages:
        Received: 1/13/2015 2:48:48 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 1
        Received: 1/13/2015 2:48:49 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 3

Connected to a VoicemailBox of an actor -3620707024940067186 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:30001/a42c9a11-9aab-4f3f-a0b8-3838
9dabba1e/c3843c54-070f-4bdc-a6f9-7f22ac062901-130656628745160706

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3620707024940067186, Greetings # 4

Playing all messages:
        Received: 1/13/2015 2:48:48 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 1
        Received: 1/13/2015 2:48:49 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 3
        Received: 1/13/2015 2:48:50 PM, Message: Hello WinFab Actor -3620707024940067186, Greetings # 4

Deleting all messages
```

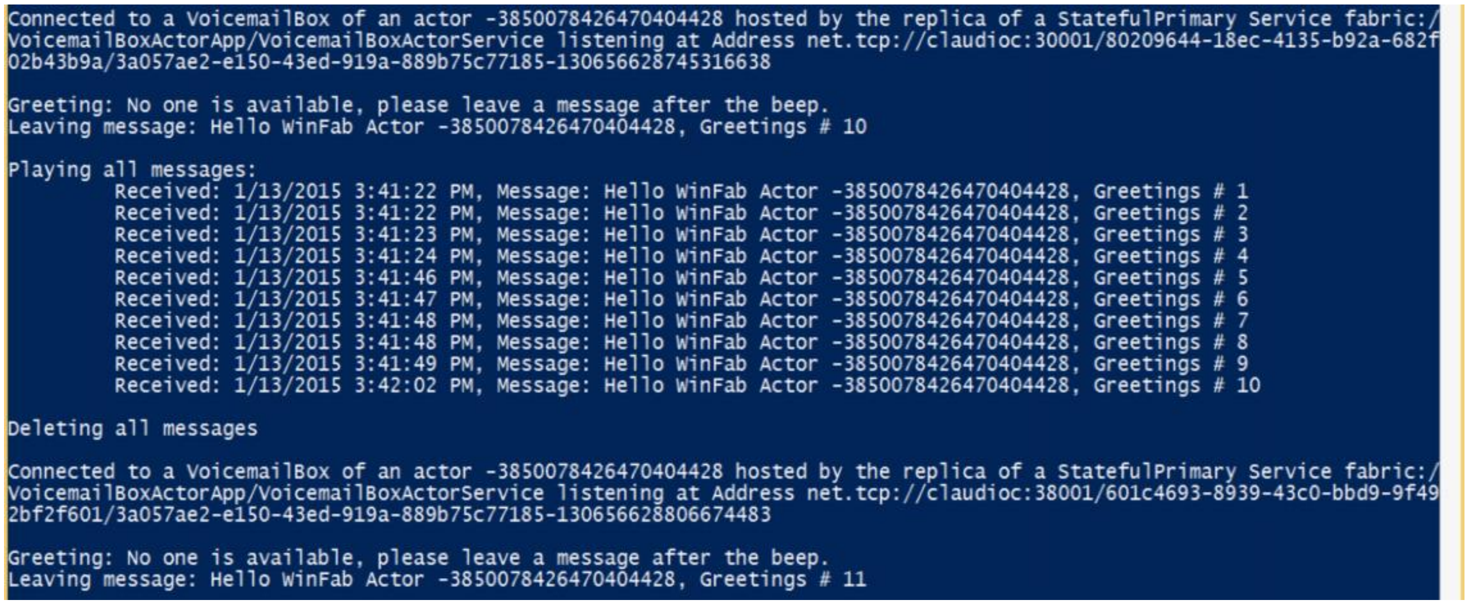
If you want you can open multiple client windows, each one would generate a new random actor id and open the same operation o n the Voicemailbox of that actor.

3. **Introduce Failover.**
   You can test the failover feature using the same process used for the Calculator sample.
   Introduce the failover by restarting the replica that hosts the VoicemailBox actor.

   ```
   .\ManageActorApp.ps1 -RestartHost –ActorType IVoicemailBoxActor -ActorId <actorId>
   ```
   (Note: The actor Id can be a negative number)

   You should see the VoicemailBox Client briefly stops for a moment, then reconnect to the failed over actor and continue. The failover may be really fast and hence you may not notice the stop in the client display. In that case you can verify that the failover has happened and the c lient is connected to the different host by looking at the address of the host it is connected to.

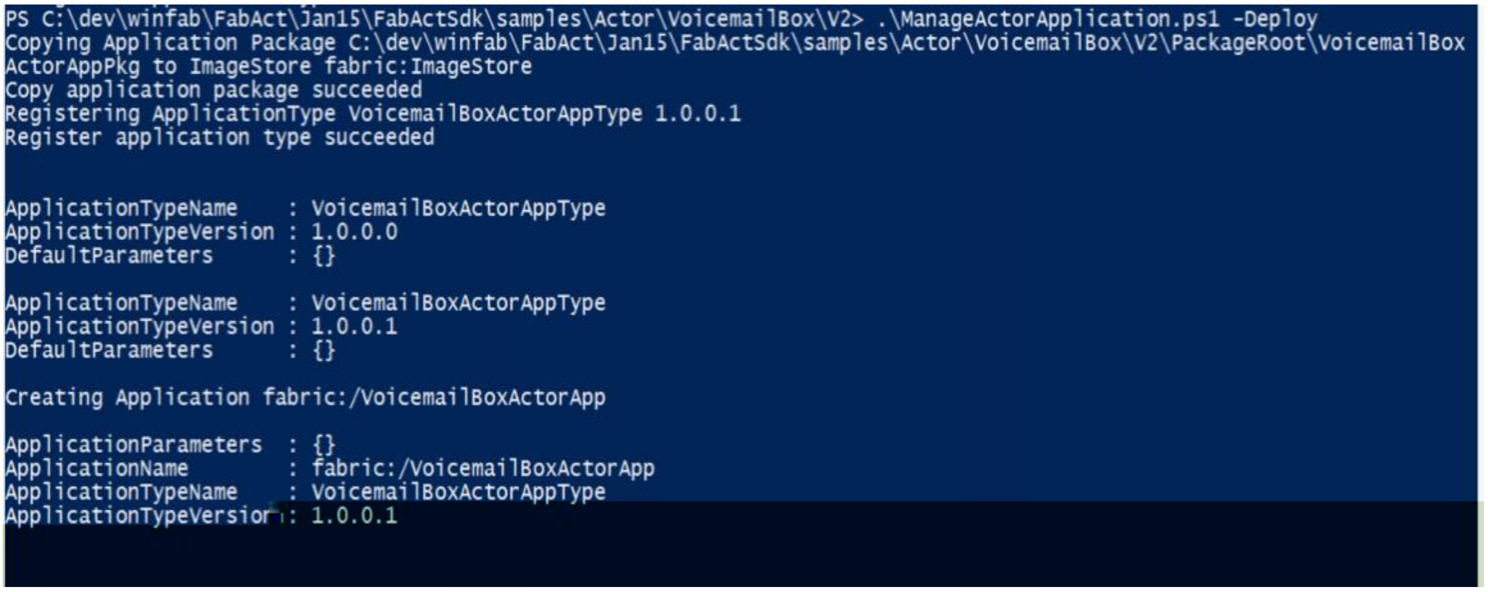
   Notice is the screenshot below that between Greeting #10 and #11 the Actor changed nodes (based on the failover triggered by the RestartHost command).



```
Connected to a VoicemailBox of an actor -3850078426470404428 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:30001/80209644-18ec-4135-b92a-682f
02b43b9a/3a057ae2-e150-43ed-919a-889b75c77185-130656628745316638

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3850078426470404428, Greetings # 10

Playing all messages:
        Received: 1/13/2015 3:41:22 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 1
        Received: 1/13/2015 3:41:22 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 2
        Received: 1/13/2015 3:41:23 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 3
        Received: 1/13/2015 3:41:24 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 4
        Received: 1/13/2015 3:41:46 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 5
        Received: 1/13/2015 3:41:47 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 6
        Received: 1/13/2015 3:41:48 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 7
        Received: 1/13/2015 3:41:48 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 8
        Received: 1/13/2015 3:41:49 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 9
        Received: 1/13/2015 3:42:02 PM, Message: Hello WinFab Actor -3850078426470404428, Greetings # 10

Deleting all messages

Connected to a VoicemailBox of an actor -3850078426470404428 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:38001/601c4693-8939-43c0-bbd9-9f49
2bf2f601/3a057ae2-e150-43ed-919a-889b75c77185-130656628806674483

Greeting: No one is available, please leave a message after the beep.
Leaving message: Hello WinFab Actor -3850078426470404428, Greetings # 11
```

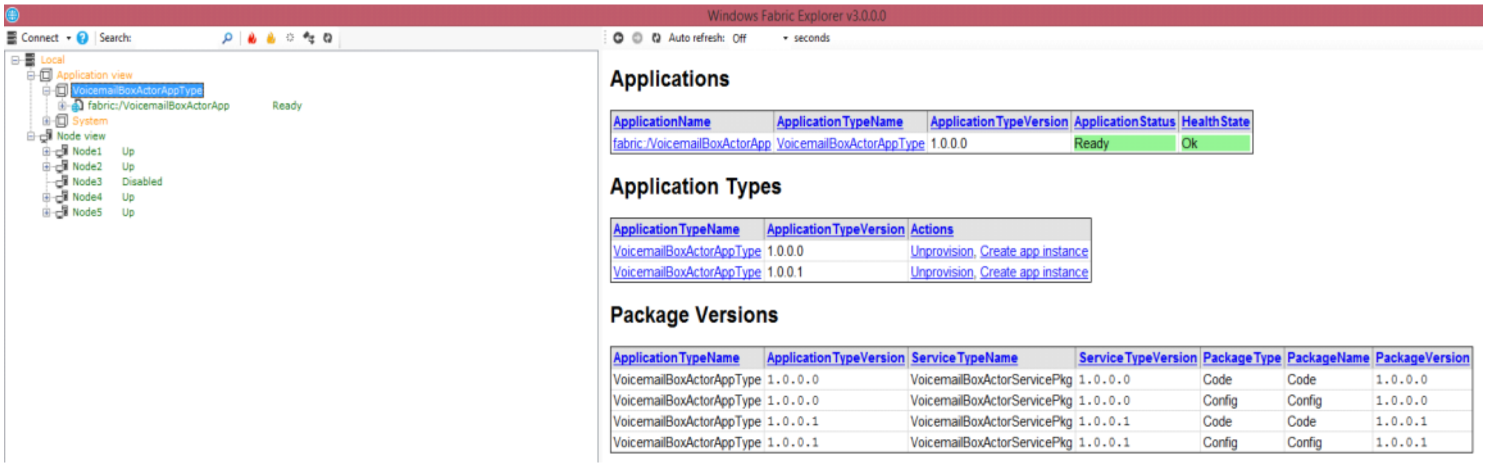
You can also introduce the failover by killing the VoicemailBoxActorService.exe on the right node. Follow the same instructi ons as described in the Calculator app sample.

1. **Upgrade the application to different version.**
   Deploy V2 of the VoicemailBoxActorService by calling the ManageActorApplication script under the V2 directory (samples \Actor\VoicemailBox\V2)



```
PS C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\V2> .\ManageActorApplication.ps1 -Deploy
Copying Application Package C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\V2\PackageRoot\VoicemailBox
ActorAppPkg to ImageStore fabric:ImageStore
Copy application package succeeded
Registering ApplicationType VoicemailBoxActorAppType 1.0.0.1
Register application type succeeded


ApplicationTypeName    : VoicemailBoxActorAppType
ApplicationTypeVersion : 1.0.0.0
DefaultParameters      : {}

ApplicationTypeName    : VoicemailBoxActorAppType
ApplicationTypeVersion : 1.0.0.1
DefaultParameters      : {}

Creating Application fabric:/VoicemailBoxActorApp

ApplicationParameters  : {}
ApplicationName        : fabric:/VoicemailBoxActorApp
ApplicationTypeName    : VoicemailBoxActorAppType
ApplicationTypeVersion : 1.0.0.1
```

Go to WinFabExplorer.exe and select the VoicemailBoxAppType. As shown in the screenshot below you should now see two versions of the application deployed to the cluster, v1.0.0.0 and v1.0.0.1



Windows Fabric Explorer v3.0.0.0

Connect · Search: · Auto refresh: Off · seconds

```
Local
  Application view
    VoicemailBoxActorAppType
      fabric:/VoicemailBoxActorApp        Ready
    System
  Node view
    Node1   Up
    Node2   Up
    Node3   Disabled
    Node4   Up
    Node5   Up
```

Applications

ApplicationName	ApplicationTypeName	ApplicationTypeVersion	ApplicationStatus	HealthState
fabric:/VoicemailBoxActorApp	VoicemailBoxActorAppType	1.0.0.0	Ready	Ok

Application Types

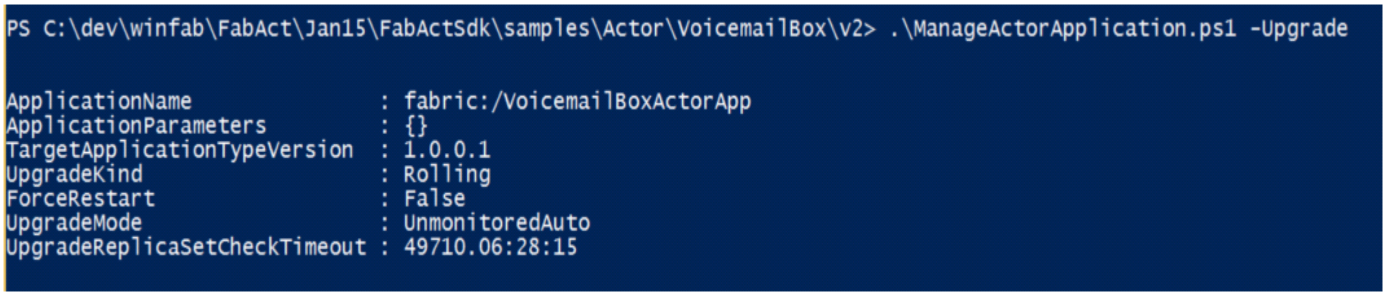
ApplicationTypeName	ApplicationTypeVersion	Actions
VoicemailBoxActorAppType	1.0.0.0	Unprovision, Create app instance
VoicemailBoxActorAppType	1.0.0.1	Unprovision, Create app instance

Package Versions

ApplicationTypeName	ApplicationTypeVersion	ServiceTypeName	ServiceTypeVersion	PackageType	PackageName	PackageVersion
VoicemailBoxActorAppType	1.0.0.0	VoicemailBoxActorServicePkg	1.0.0.0	Code	Code	1.0.0.0
VoicemailBoxActorAppType	1.0.0.0	VoicemailBoxActorServicePkg	1.0.0.0	Config	Config	1.0.0.0
VoicemailBoxActorAppType	1.0.0.1	VoicemailBoxActorServicePkg	1.0.0.1	Code	Code	1.0.0.1
VoicemailBoxActorAppType	1.0.0.1	VoicemailBoxActorServicePkg	1.0.0.1	Config	Config	1.0.0.1

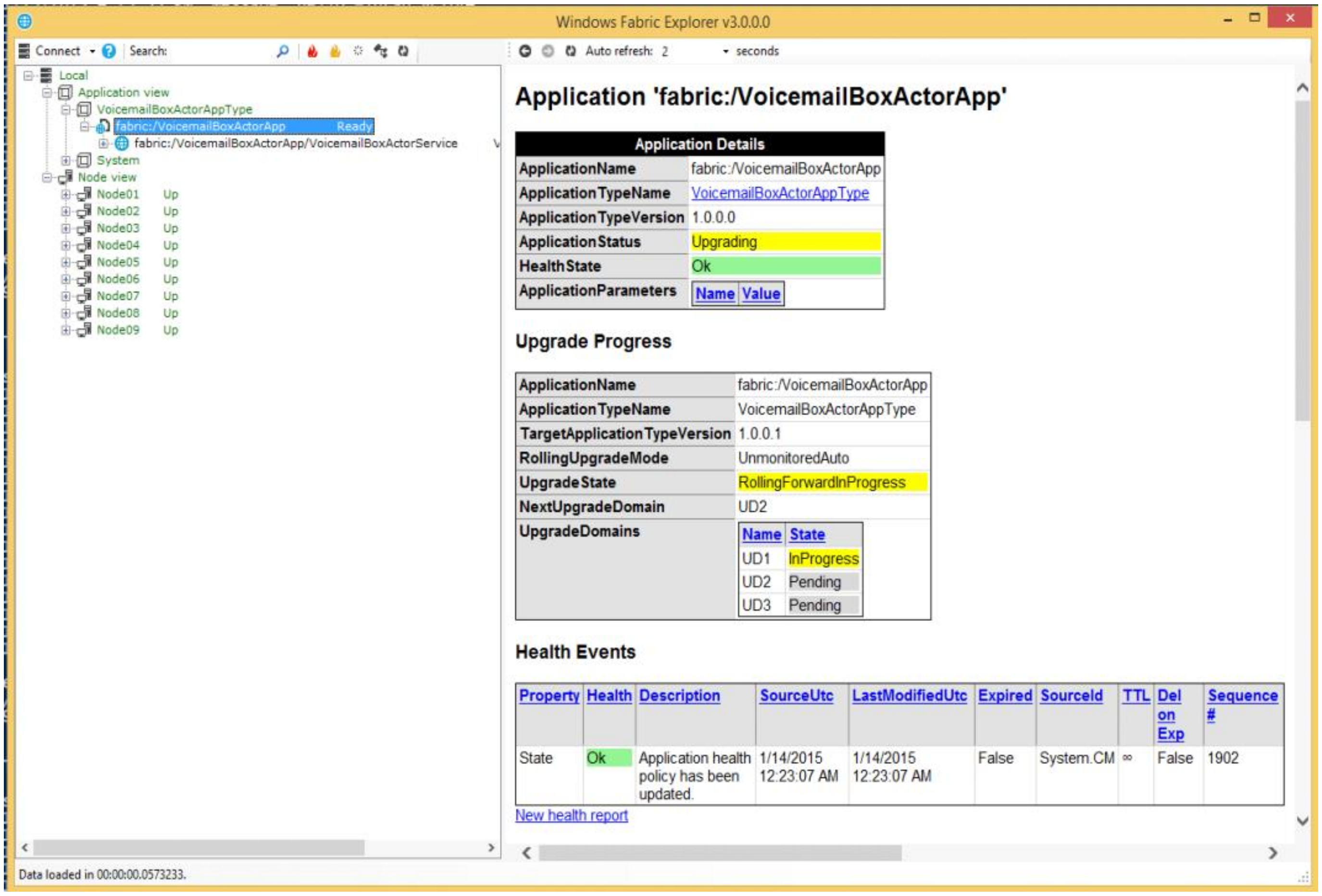
With the Client running perform the upgrade after deploying the new version. Run the following powershell command.

```
.\ManageActorApplication.ps1 –Upgrade
```



```
PS C:\dev\winfab\FabAct\Jan15\FabActSdk\samples\Actor\VoicemailBox\v2> .\ManageActorApplication.ps1 -Upgrade


ApplicationName                : fabric:/VoicemailBoxActorApp
ApplicationParameters          : {}
TargetApplicationTypeVersion   : 1.0.0.1
UpgradeKind                    : Rolling
ForceRestart                   : False
UpgradeMode                    : UnmonitoredAuto
UpgradeReplicaSetCheckTimeout  : 49710.06:28:15
```

Upgrade should finish in few minutes. You can monitor the progress of the upgrade from WinFabExplorer. Click on the Applicati on and put WinFabExplorer in the auto-refresh mode. See the picture below. While the upgrade is going on you will notice that the client continues to connect to th e service and work.



Windows Fabric Explorer v3.0.0.0

Connect | Search: | Auto refresh: 2 seconds

```
Local
  Application view
    VoicemailBoxActorAppType
      fabric:/VoicemailBoxActorApp          Ready
        fabric:/VoicemailBoxActorApp/VoicemailBoxActorService
    System
  Node view
    Node01    Up
    Node02    Up
    Node03    Up
    Node04    Up
    Node05    Up
    Node06    Up
    Node07    Up
    Node08    Up
    Node09    Up
```

Application 'fabric:/VoicemailBoxActorApp'

Application Details	
ApplicationName	fabric:/VoicemailBoxActorApp
ApplicationTypeName	VoicemailBoxActorAppType
ApplicationTypeVersion	1.0.0.0
ApplicationStatus	Upgrading
HealthState	Ok
ApplicationParameters	Name | Value

Upgrade Progress

ApplicationName	fabric:/VoicemailBoxActorApp
ApplicationTypeName	VoicemailBoxActorAppType
TargetApplicationTypeVersion	1.0.0.1
RollingUpgradeMode	UnmonitoredAuto
UpgradeState	RollingForwardInProgress
NextUpgradeDomain	UD2
UpgradeDomains	

Name	State
UD1	InProgress
UD2	Pending
UD3	Pending

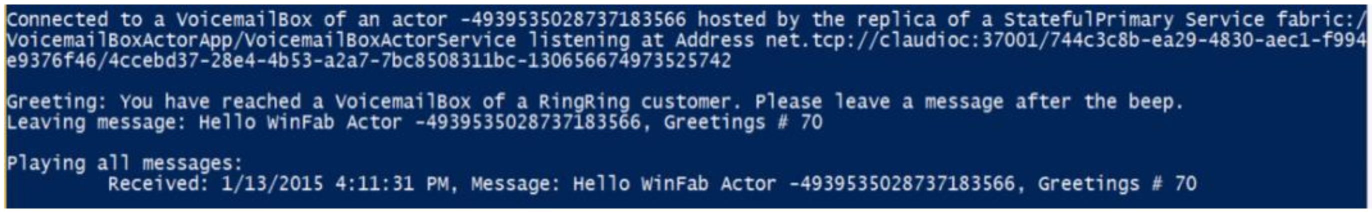
Health Events

Property	Health	Description	SourceUtc	LastModifiedUtc	Expired	SourceId	TTL	Del on Exp	Sequence #
State	Ok	Application health policy has been updated.	1/14/2015 12:23:07 AM	1/14/2015 12:23:07 AM	False	System.CM	∞	False	1902

New health report

Data loaded in 00:00:00.0573233.

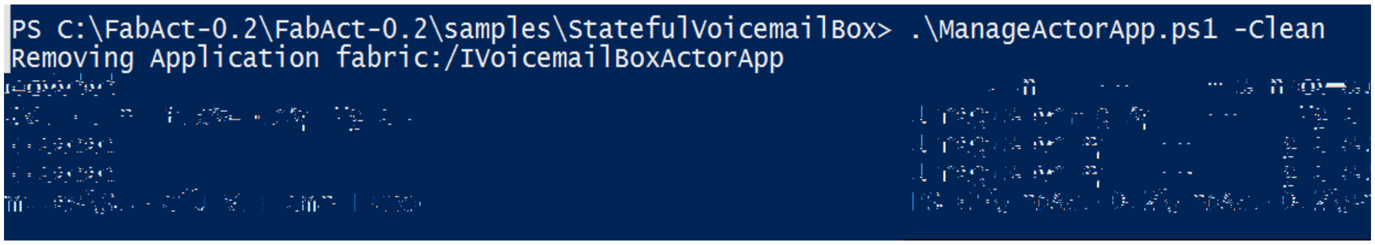
At this time, you should see the default greeting for the VoicemailBox change to
“You have reached a VoicemailBox of a RingRing customer”.



```
Connected to a VoicemailBox of an actor -4939535028737183566 hosted by the replica of a StatefulPrimary Service fabric:/
VoicemailBoxActorApp/VoicemailBoxActorService listening at Address net.tcp://claudioc:37001/744c3c8b-ea29-4830-aec1-f994
e9376f46/4ccebd37-28e4-4b53-a2a7-7bc8508311bc-130656674973525742

Greeting: You have reached a VoicemailBox of a RingRing customer. Please leave a message after the beep.
Leaving message: Hello WinFab Actor -4939535028737183566, Greetings # 70

Playing all messages:
        Received: 1/13/2015 4:11:31 PM, Message: Hello WinFab Actor -4939535028737183566, Greetings # 70
```

1. **Remove the deployment from the cluster.**
   In order to reset the demonstration, you should clean the deployment of the VoicemailBox Actor from the cluster by calling th e following command. WARNING: Before you do this, make sure that WinFabExplorer is no longer in the Auto Refresh mode, otherwise it will continue to show e rror dialogs. If that happens, kill WinFabExplorer using task manager.

   `.\ManageActorApp.ps1 -Clean`



```
PS C:\FabAct-0.2\FabAct-0.2\samples\StatefulVoicemailBox> .\ManageActorApp.ps1 -Clean
Removing Application fabric:/IVoicemailBoxActorApp
[illegible]
```

-