# Microsoft Dynamics CRM SDK

## AppFabric Queued Listener Readme

## This folder contains sample code demonstrating how to write a Windows Azure AppFabric Listener and connect it to Microsoft Dynamics CRM 2013.

## Instructions

## AppFabric Queued Listener Project

## Sign up for Windows Azure Platform

1. Navigate to <http://www.microsoft.com/windowsazure/> and follow the steps to sign up for a Windows Azure account. As of this writing setting up an account requires a credit card but there are trial accounts which at the time of this writing do not charge the credit card used. Please read any agreements entered into carefully as the details may have changed.
2. Configure Windows Azure AppFabric ACS for Microsoft Dynamics CRM integration by following the instructions in the topic [Configure Windows Azure Integration with Microsoft Dynamics CRM](http://msdn.microsoft.com/en-us/library/gg309340.aspx).

## Add Windows Azure Access Rights for the Microsoft Dynamics CRM Server

1. From the server hosting Microsoft Dynamics CRM
2. Open a PowerShell Command Prompt.
3. Type ‘Add-PSSnapin Microsoft.Crm.Powershell’
4. Type ‘Get-CrmCertificate’
   1. If there are no certificates one must be added before proceeding using Set-CrmCertificate
5. Note the name of the certificate as this needs to match an issuer’s issuername in Windows Azure.
6. Open the Start Menu select ‘Run…’
7. Enter ‘certmgr.msc’ and press enter to open Certificates
8. Expand the ‘Trusted Root Certification Authorities’ node
9. Click on the ‘Certificates’ node
10. Right-click a certificate issued by the localhost
11. From the context menu Press ‘All Tasks’
12. Press ‘Export’
13. Follow the export wizard to export a base-64 encoded .cer file to a machine with the Windows Azure Platform AppFabric SDK installed
14. From the machine with the Windows Azure Platform AppFabric SDK installed open a command prompt
15. In the command prompt Navigate to the Windows Azure Platform AppFabric SDK tools directory (such as %ProgramFiles%\Windows Azure platform AppFabric SDK\V1.0\Tools
16. In this directory edit the acm.exe.config
    1. Set the value for the key ‘service’ equal to the Windows Azure AppFabric service namespace.
    2. Set the value for the key ‘mgmtkey’ equal to the Windows Azure AppFabric Issuer Key
17. Type ‘Acm create issuer –name:CrmEndpoint –issuername:ISSUERNAMEFROMABOVE –certfile:LOCATIONOFTHEEXPORTEDCERTFILE –algorithm:X509’ and press enter.
18. On success, ‘Object created successfully’ with an identifier in parentheses should print to the screen this value will be supplied for ISSUERIDFROMACMCREATEISSUER when issuing the acm create rule command.
19. Type ‘acm getall scope’ there should be one scope listed. The id for it will be supplied for SCOPEIDFROMGETALLSCOPE in the acm create rule command.
20. Type ‘acm create rule -name:CrmSend -scopeid:SCOPEIDFROMGETALLSCOPE -inclaimissuerid:ISSUERIDFROMACMCREATEISSUER -inclaimtype:Issuer -inclaimvalue:localhost -outclaimtype:net.windows.servicebus.action -outclaimvalue:Send
21. On Success ‘Object created successfully’ with an identifier in parentheses should print to the screen.
22. Please refer to the Microsoft Dynamics CRM 2013 SDK topic, Configure Windows Azure Integration with Microsoft Dynamics CRM, for more information.

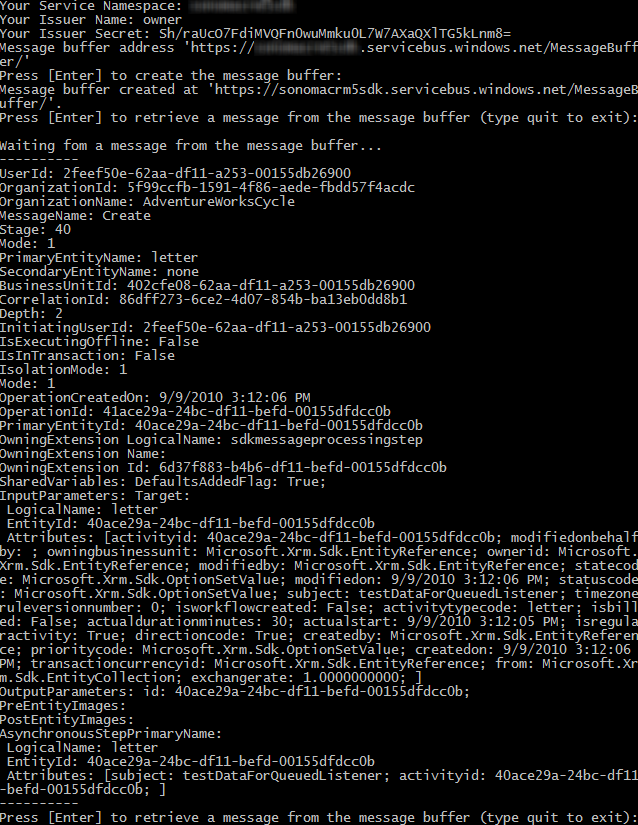
## Configuring Microsoft Dynamics CRM to post to Windows Azure AppFabric on creation of a letter.

1. Using the PluginRegistration Tool in the Microsoft Dynamics CRM SDK, connect to the Microsoft Dynamics CRM organization which will post to the Windows Azure AppFabric endpoint.
2. Press the ‘Register’ button and from the dropdown list select ‘Register New Service Endpoint’
3. In the Name field enter a meaningful identifier such as ‘QueuedListener’
4. In the Description field enter ‘AppFabric Queued sample’
5. In the Solution Namespace field enter YOUR AZURE SERVICE NAMESPACE
6. In the Path field enter ‘MessageBuffer’
7. In the Contract dropdown select ‘Queue’
8. Ensure the Claim dropdown is set to ‘None’
9. Press the ‘Save’ button to save the Endpoint configuration.
10. Under the ‘Registered Plugins & Custom Workflow Activities list right click the ‘(ServiceEndpoint) QueuedListener’ (the identifier from step #3) node.
11. Select ‘Register New Step’
12. In the Message field enter ‘Create’
13. In the Primary Entity field enter ‘letter’
14. Press ‘Register New Step’
15. Plugin Registration completed successfully. Close the tool.

## How to Install, Build, and Run the Sample

1. In Windows Explorer, launch Microsoft Visual Studio 2010.
2. Open the WindowsAzureVB solution.
3. Right-click on the QueuedListener Project.
4. Select the ‘Set as StartUp Project’ menu option.
5. Compile and run the project by pressing F5.
6. At the prompt, enter appropriate information (as shown in the example screenshot below) to start the listener.
7. When the console app displays ‘Press [Enter] to retrieve a message from the message buffer (type quit to exit) create and save a new letter activity in Microsoft Dynamics CRM and then press enter to view the message.

**Expected Results**

You should see:  


## Troubleshooting

## Build error(s)

## Ensure the proper references are added and the project is targeting the .NET Framework 4.0 and not the .NET Framework 4 Client Profile.

## Run-time errors

## If when running the project it hangs while the host is opening check if Fiddler or some other proxy is preventing the sample from working.

## If everything seems to run fine but upon creating a new letter nothing shows up, open Microsoft Dynamics CRM, navigate to Settings, System Jobs, and look for a job regarding the created letter with ‘Waiting’ for Status Reason. Open that job and view details to get troubleshooting information.

## If you are experiencing errors when running the samples, check the following.

## You have followed all instructions in this document correctly.

## The server, organization, and logon credentials you entered are correct.

## Your account on the CRM server has the required privileges to perform the sample’s intended operation.

## Important Notes

* The sample files should not be used in a production environment without prior testing. You should deploy an application that uses this sample code to a test environment and examine it for interaction or interference with other parts of the system.
* Before you deploy applications that use this sample code to a production environment, make sure that you consider the existing customizations that you may have implemented in Microsoft Dynamics CRM.

## Copyright

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, Active Directory, Outlook, Visual Studio, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.