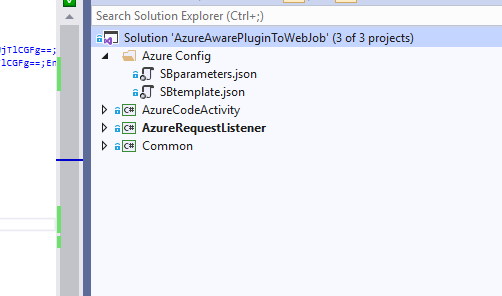
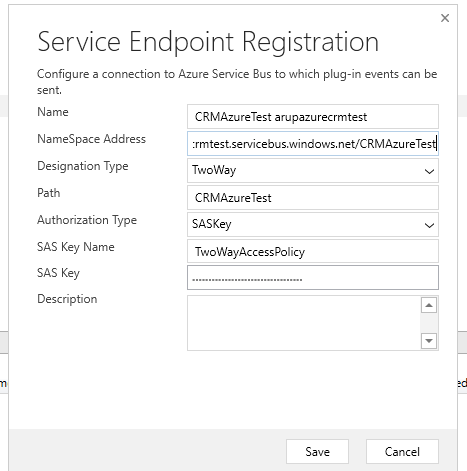
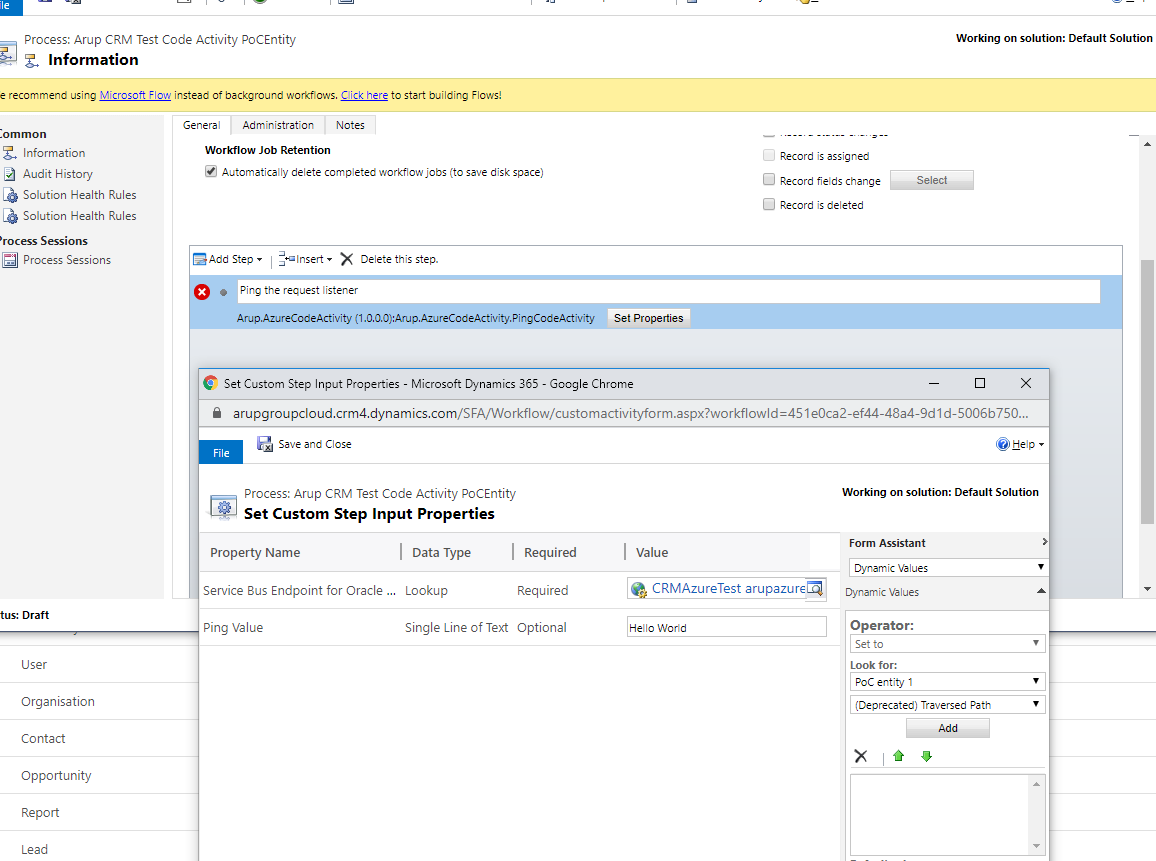
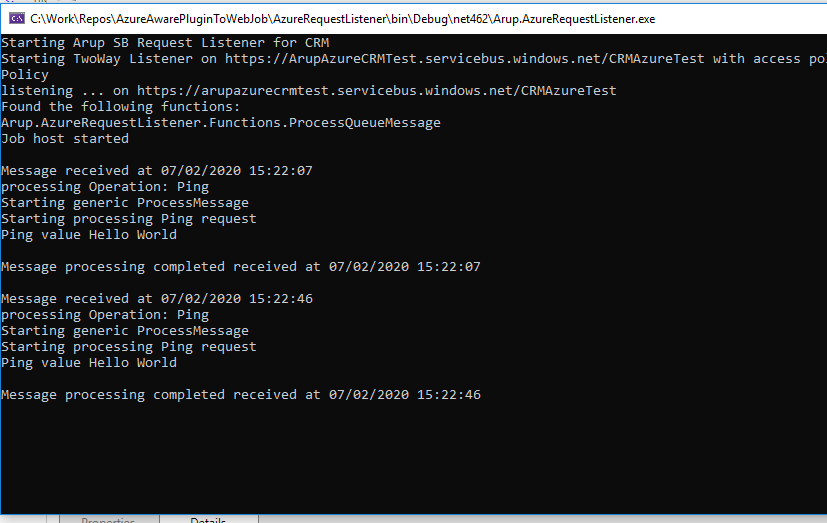
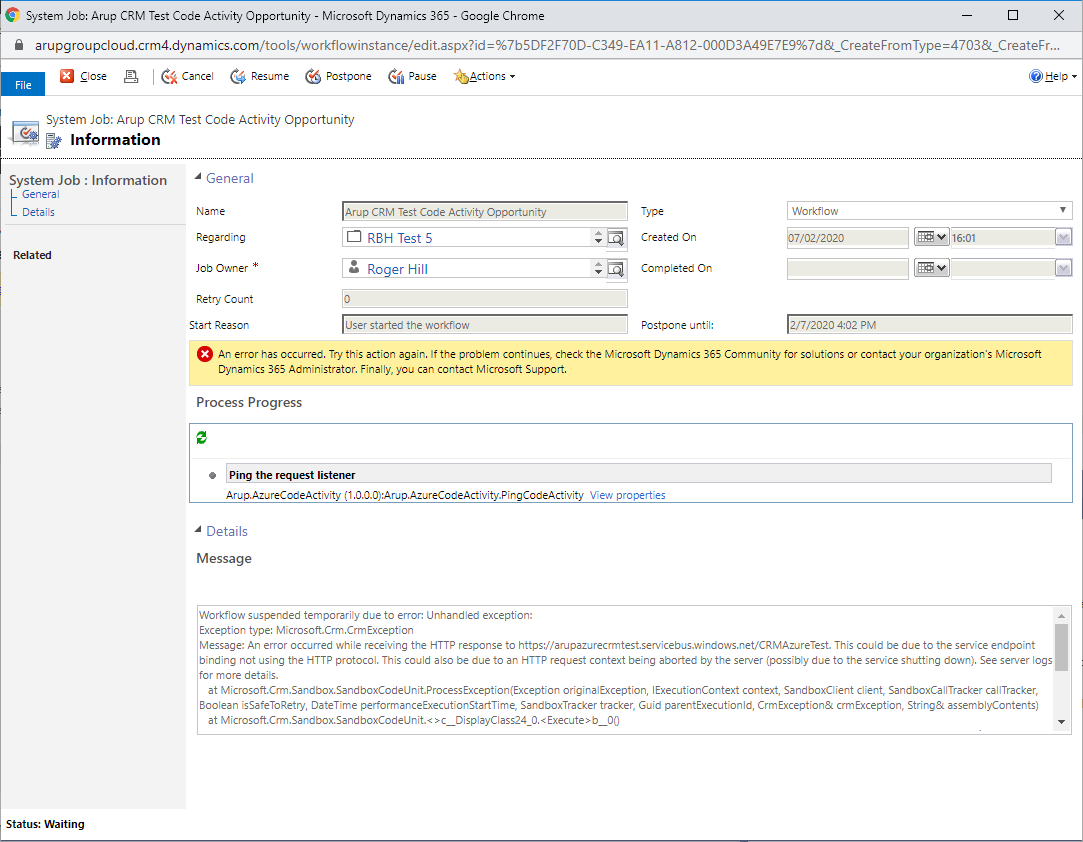
# AzureAwarePluginToWebjob

The source for the demo solution is stored in Githut.

1. Clone the repository from <https://github.com/rogerhillgsy/AzureAwarePluginToWebJob>
2. Open the AzureAwarePluginToWebJob solution and build.  
   
3. Create an Azure Service bus. (i.e. <https://ArupAzureCRMTest.servicebus.windows.net/>)
4. On the service bus, create a shared access policy “TwoWayAccessPolicy” with Send+Listen access.
5. Configure the app.config file of the AzureRequestListener with values for ServiceBusSASKey, SharedAccessKyeName and ServiceBusEndpoint. (e.g <https://ArupAzureCRMTest.servicebus.windows.net/CRMAzureTest/> )  
   Note that the exact path (/CRMAzureTest) is not important, but it has to be consistent with the service endpoint configuration created in CRM.
6. The AzxureRequestListener can be run directly from Visual Studio (and I would recommend this)
7. An alternative is to create an Azure App Service and publish the “AzureRequestListener” as a WebJob to the appservice (make sure it is set to run continuously)
8. Open the Plugin Registration Tool. Copy the Servicebus SAS key connection string for the TwoWayAccessPolicy, and use this to set up the set up a new service endpoint.   
   (Note – to avoid errors, you will need to change the “Designation Type” field to TwoWay before changing the scheme of the NameSpaceAccess from sb: to https: ) See screenshot below.  
   
9. Use the plugin registration tool to deploy the code activity assembly (Arup.AzureCodeActivity.dll) to CRM.
10. Create a workflow to run the “PingCodeActivity” (see screenshot below.)   
    Add a PingCodeActivity step to the workflow.  
    Configure the Ping step with the identity of the ServiceEndpoint, and choose a ping value.  
    
11. If the workflow is run on an entity created in the online instance (in my case rbh\_pocentityforazureplugins) then it runs successfully, producing the output shown here: -



1. If it is run on an entity migrated from our on premise CRM2016 instance, such as Opportunity, it fails.  
   I have tried this with other entities with similar results.



The full error message is shown below.

|  |
| --- |
| Workflow suspended temporarily due to error: Unhandled exception:  Exception type: Microsoft.Crm.CrmException  Message: An error occurred while receiving the HTTP response to https://arupazurecrmtest.servicebus.windows.net/CRMAzureTest. This could be due to the service endpoint binding not using the HTTP protocol. This could also be due to an HTTP request context being aborted by the server (possibly due to the service shutting down). See server logs for more details.  at Microsoft.Crm.Sandbox.SandboxCodeUnit.ProcessException(Exception originalException, IExecutionContext context, SandboxClient client, SandboxCallTracker callTracker, Boolean isSafeToRetry, DateTime performanceExecutionStartTime, SandboxTracker tracker, Guid parentExecutionId, CrmException& crmException, String& assemblyContents)  at Microsoft.Crm.Sandbox.SandboxCodeUnit.<>c\_\_DisplayClass24\_0.<Execute>b\_\_0()  at Microsoft.PowerApps.CoreFramework.ActivityLoggerExtensions.Execute(ILogger logger, EventId eventId, ActivityType activityType, Action action, IEnumerable`1 additionalCustomProperties)  at Microsoft.Xrm.Telemetry.XrmTelemetryExtensions.Execute(ILogger logger, XrmTelemetryActivityType activityType, Action action)  at Microsoft.Xrm.RemotePlugin.CrmProvider.RemotePlugin.CDSExecute(IServiceProvider serviceProvider)  at Microsoft.Crm.Workflow.Services.ProxyCustomActivity.Execute(CodeActivityContext executionContext)  -- End stack trace -- |