SMVCloudWorker – Overview

# Inputs

The worker takes input from four sources:

* Actions Queue: The cloud storage queue smvactions is read for messages. Each message in the queue represents an action to be run by a worker in the cloud service.
* Actions blob container: The smvactions blob container contains the entire working directory for the action zipped up into an archive named {actionGuid}.zip
* Versions blob container: Each action is run by a specific version of SMV. The versions are downloaded by the worker from the smvversions blob container when required and are persisted to disk until the worker is reimaged.
* Actions table: The cloud table actionstable contains an entry for each action added to the actions queue. This table primarily serves five purposes:
  + To inform the client about the status of the action (NotStarted, InProgress, Success, Error).
  + To allow the worker access to the action object that needs to be executed via the SerializedAction property. Once the action has been executed, the SerializedAction property is updated with the new variables dictionary for the action so the client can access variables that were set during the action’s execution.
  + The ModuleHash property is used when an action needs to download artifacts from a module during execution.
  + The PluginPath property is used by the worker to load the plugin so we can run PreAction() and PostAction() during execution.
  + The Version property is used by the worker to download the version of SMV that will be used for executing this action.

# Outputs

After an action has finished execution, the worker will:

* Upload an archive named {actionGuid}.zip to the smvresults blob container. The archive will contain the contents of the working directory after the action has completed execution.
* Set the status of the action in the table to Success.
* Send a message to the Service Bus topic smvresults with the following properties:
  + SchedulerInstanceGuid – A GUID assigned to each instance of the CloudActionSMVScheduler.
  + ActionGuid – A GUID assigned to each action by the client.
  + DequeueCount – The number of times the message has been dequeued so far. This is essentially the number of times the action has failed to run.
  + WaitTime – The time difference between the time the message was inserted into the queue and when the message was picked up by the worker.

# Working

1. Read a message from the actions queue. Messages are of the form: {schedulerInstanceGuid},{actionGuid}
2. Get the table entry for this action from the actions table.
3. Check the dequeue count to find out how many times we have attempted to process this message. If the dequeue count is greater than MaxDequeueCount, we will discard the message and report an error to the client.
4. Switch to the version of SMV required for the action by looking at the Version property of the table entry. If the SMV version does not exist on disk, we download it from blob storage and switch to it.
5. Load the plugin, if any, by using the PluginPath property in the table entry.
6. Load the module, if any, by using the ModuleHash property of the table entry.
7. Download and extract the job from blob storage.
8. Before running the action, we transform all paths in the variables dictionary by replacing the working directory and assembly directory in the client with the working directory and assembly directory in the worker.
9. Call Utility.ExecuteAction() to run the action.
10. Package the working directory and upload it to the results blob container.
11. Set the status of the action to complete and update the serialized action in the table.
12. Send a message to our service bus topic to inform the client that the action is complete.
13. If there an exception occurred during this process, log the error message, make the message in the actions queue visible again and go to step 1.
14. Go to step 1.