**MQInput, MQOutput and MQGet node properties change**

**Manual steps involved in updating bar file for MQ related nodes**

Below are the steps involved in manual update of bar file/properties file for an application which involves MQ local binding. All the MQ related nodes needs to be updated with client connection information to deploy the application as containers in CP4I environment.

Step 1:

Identify QM details:

1. Host name = <host name of QM instance>
2. Listener Port number = <Port number of QM instance, default 1414>
3. Communication Channel name = <Server-Connection channel name>
4. Destination QM name
5. Security identity (Optional)
6. Queue name

Step 2:

Paste the bar file into tool kit and update the bar file with the above values (all the MQ related nodes)

Step 3:

Save the bar file

Step 4:

Generate the new properties file for future reference

Step 6:

Run TA for the new bar file

Step 7:

Deploy the bar file

The above steps looks simple but needs to be done for all bar files and all the MQ nodes involved in it. Which will eventually a tedious repeated process and bound to result in human error. Approximately 20 minutes is needed to update one bar file (depends on number of MQ nodes to be updated)

**Proposed Automation Steps**

There are two process involved in this automation.

1. To read TA Output json file and generate output with the node names for which MQ client connection information needs to be updated

2. To read QM data for client connection information and update the properties file automatically

Step 1:

Take the Transformation advisor output JSON file.

Sample TA output JSON file:



Step 2:

If “title” element in json contains the message as below:

"title": "An MQInput, MQOutput, or MQGet message flow node using server bindings to a queue manager has been found. You might want to consider changing this when moving to containers."

Then MQ related nodes needs to be updated to client server binding while moving the container

Step 3:

Identify the flow name and impacted node names from the recommendation json

Ex:

Application name = "name": "AggregateMQFlow"

Using OccurrencesCount field traverse through occurrences array

Ex:

"occurrencesCount": 5

"occurrences": [

{

"app": "AggregationMQ",

"sharedLibName": "",

"staticLibName": "",

"subFlow": "",

"messageFlow": "AggregationMQ\_FanIn",

"flowNode": "MQ Input - BACKEND.REPLY"

},

{

"app": "AggregationMQ",

"sharedLibName": "",

"staticLibName": "",

"subFlow": "",

"messageFlow": "AggregationMQ\_FanIn",

"flowNode": "MQ Output - AGG.OUT"

},

{

"app": "AggregationMQ",

"sharedLibName": "",

"staticLibName": "",

"subFlow": "",

"messageFlow": "AggregationMQ\_FanOut",

"flowNode": "MQ Input - AGG.IN"

},

{

"app": "AggregationMQ",

"sharedLibName": "",

"staticLibName": "",

"subFlow": "",

"messageFlow": "AggregationMQ\_FanOut",

"flowNode": "MQ Output - BACKEND1"

},

{

"app": "AggregationMQ",

"sharedLibName": "",

"staticLibName": "",

"subFlow": "",

"messageFlow": "AggregationMQ\_FanOut",

"flowNode": "MQ Output - BACKEND2"

}

]

**Step 4:**

Based on JSON array (occurrences array) from above step, get messageFlow field value and corresponding flowNode field value

Ex:

"messageFlow": "AggregationMQ\_FanOut",

"flowNode": "MQ Output - BACKEND2"

"messageFlow": "AggregationMQ\_FanOut",

"flowNode": "MQ Output - BACKEND2"

**Step 5:**

For each flow Node (MQ nodes) below field values are needed as input:

1. Host name = <host name of QM instance>
2. Listener Port number = <Port number of QM instance, default 1414>
3. Communication Channel name = <Server-Connection channel name>
4. Destination QM name (Optional)
5. Security identity (Optional)
6. Queue name (Optional)

Apart from above node specific inputs, user has to provide the below data too

1. What kind of flow? Application or Library or Independent Message Broker Project or REST Interface
2. Bar file location – folder ( complete path with file name) at which the bar file is available (which needs to be updated)
3. Properties file location – folder(complete path with file name) at which the properties file is available (which needs to be updated)
4. Region or environment ? Dev/QA/Stage/Prod – which regions or environment’s properties file needs to be updated

Sample input for the tool:



**Step 6:**

This will update the properties file with the data necessary to override the bar file with client connection channel which is supported in cp4i environment. The updated properties file can be used to override the bar file and ready to deploy into cp4i region.

All the above steps except Step 5 are automated. The values mentioned in Step 5 are the needed input values for the tool to automatically update the properties file with client connection information for MQ nodes.

Please refer the MQPropertiesAutomation attachment in JumpStartPortal -> BestPractices -> Refactoring section