

IBM Integration Bus – MQInput Generic Error Handler Subflow

Sudhakar Bodapati

March 3rd 2019

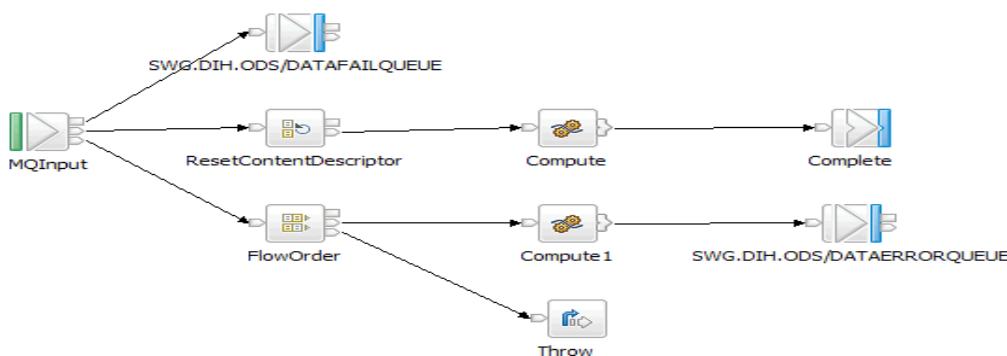
Introduction

This is Generic Error Handling Subflow framework for IBM Integration Bus to simplify exception handling in a message flow that uses MQInput node. Below are the benefits of using MQ Generic Error Handling framework.

- A. It's fully functional as the Typical Error Handling Style
- B. Many functions included in MQ/IIB Generic Error Handler
 - a. Profiling the Error (System Error, Data Error)
 - b. Retry Logic (Automatic and manual retry)
 - c. Notification function (i.e. Email when needed)
 - d. Stop the flow when error happens
 - e. Auditing is enabled to log incoming, outgoing messages.
- C. Reduces development time
 - a. Can be very easily plugged by the developers
 - b. All the parameters are configurable
- D. Reduces Administration time
 - a. Standard place to look for errors
 - b. Standard reprocessing logic (Retry, Stop Message Flow)

Typical Error Handling Style

- A. Error Happens in downstream nodes
- B. Catch Terminal to catch exceptions
- C. Failure Terminal



After plugging in the MQInput Generic Error Handler



Patterns

- 1) Write to ERROR_QUEUE when Any error
- 2) Stop flow when Any error
- 3) Stop flow when a System error
- 4) Retry when Any error
- 5) Retry when System error
- 6) Retry and Stop flow when Any error
- 7) Retry and Stop flow when System error

Configuration Files

The configuration files are optional, but use it depending the pattern you select.

FILES	DESCRIPTION
env.dat	<p>File contains environment name which will be used in gehs_source_target_interface.xml file.</p> <p>Filename: <code>env.dat</code> ENVIRONMENT=Development – will be used in EmailSubject.</p>
GEHS_CONFIG_FILEPATH	<p>Contains parameters for Advanced, and EMail Tabs. Parameters defined here will override the message flow settings. Useful to change settings at runtime without making changes at flow level.</p> <p>Filename: <code>gehs_source_target_interfacename.xml</code></p> <pre> <?xml version="1.0" encoding="UTF-8"?> <GEHS> <ENV_FILENAME>C:\home\mqmswg\work\env.dat</ENV_FILENAME> <Advanced> <STOP THESE FLOWS></STOP THESE FLOWS> <SYSTEM_ERROR_CODES></SYSTEM_ERROR_CODES> <RETRY> <RETRY_INTERVAL_IN_MINUTES></RETRY_INTERVAL_IN_MINUTES> <RETRY_COUNT></RETRY_COUNT> </RETRY> </Advanced> <EMail> <EMAIL_FROM></EMAIL_FROM> <EMAIL_TO></EMAIL_TO> <EMAIL_CC></EMAIL_CC> <EMAIL_BCC></EMAIL_BCC> <EMAIL SUBJECT></EMAIL SUBJECT> <EMAIL_SERVER_AND_PORT> </EMAIL_SERVER_AND_PORT> <EMAIL_ATTACHMENT_FILENAME></EMAIL_ATTACHMENT_FILENAME> </EMail> </GEHS> </pre>
AUDIT_CONFIG_FILEPATH	<p>Contains parameters for the Audit, to turn on/off logging and change message expiry.</p> <p>Filename: <code>audit.xml</code></p> <pre> <Audit> <cache_refresh_interval>60</cache_refresh_interval> <!--Interface details--> </pre>

```

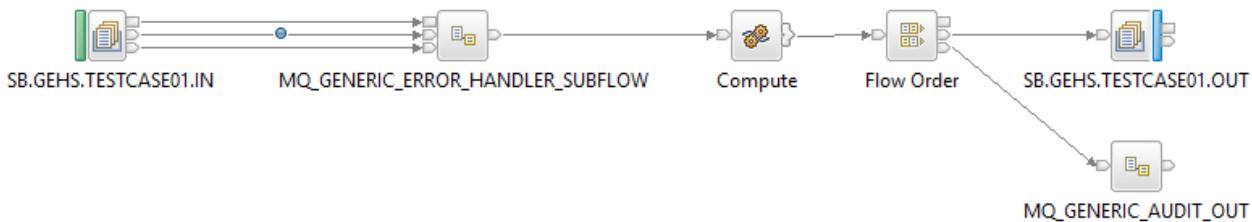
<INTERACE-NAME-Y>          /* Your interface name */
  <write_to_queue>Y</write_to_queue>
  <message_expiry_interval>10080</message_expiry_interval>
</INTERACE-NAME-Y>
<INTERACE-NAME-X>          /* Your interface name */
  <write_to_queue>Y</write_to_queue>
  <message_expiry_interval>10080</message_expiry_interval> -- MINUTES , 10080 - 7
DAYS
</INTERACE-NAME-X>
</Audit>

```

Patterns

1. Write to **ERROR_QUEUE** when Any error

Use this scenario to output failed messages to **ERROR_QUEUE**. The exception reason will be recorded into **mqrhf2.usr** folder.



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	Queue name: SB.GEHS.TESTCASE01.ERROR
Advanced	Connection*: SERVER
EMail	Destination queue manager name: <Integration node queue manager>
Audit_IN	Queue manager host name: Listener port number: Channel name: Security identity: Use SSL: <input type="checkbox"/> SSL peer name: SSL cipher specification:

Image: Error Queue details above

Expected Result:

- If any error in the main flow logic the message will be put into **ERROR_QUEUE**, the exception will be logged onto the **mqrhf2.usr** section. In the above scenario, the output queue **GEHS.TESTCASE01.OUT** is full, hence the flow will fail and the message will be routed to the error queue.
- If the **ERROR_QUEUE** is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.
- The exception details will be set to **mqrhf2.usr** section as below,


```
<gehs><pid>7876</pid><broker>TESTNODE_superuser</broker><exgrp>GEHS</exgrp><messageflow>TC1_ALL_ATTRIBUTES_ARE_OPTIONAL</messageflow><error>2666-Failed to open queue MQW101
```

```

2085
IB10QMGR
GEHS.TESTCASE01.OUT
2</error><resubmitTo>GEHS.TESTCASE01.IN</resubmitTo><mqmdorgformat>MQSTR
</mqmdorgformat><mqmdorgccsid>437</mqmdorgccsid><mqmdorgencoding>546</mqmdorgencoding></gehs>

```

2. Stop flow when Any error

Use this scenario to stop message flow(s) when any error in the message flow.

Set BOTHRESH, and BOQNAME parameters of the Input Queue.

Set BOQNAME to be same as the Input Queue name.

Input Queue Definition,

DEF QLOCAL (SB.GEHS.TESTCASE02.IN) BOQNAME(SB.GEHS.TESTCASE02.IN) BOTHRESH(100)

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	Queue name: SB.GEHS.TESTCASE02.ERROR Connection*: SERVER Destination queue manager name: <Integration node queue manager> Queue manager host name: Listener port number: Channel name: Security identity: Use SSL: SSL peer name: SSL cipher specification:
Advanced	
EMail	
Audit_IN	

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	GEHS_CONFIG_FILEPATH: /home/mesadmin/sudhakar/GEHS_TC2_STOP_FLOW_WHEN_ANY_ERROR.xml
Advanced	STOP_THESE_FLOWS: <input checked="" type="checkbox"/> STOP_FLOW_WHEN_ANY_ERROR: <input checked="" type="checkbox"/> RETRY_WHEN_ANY_ERROR: <input type="checkbox"/> STOP_FLOW_WHEN_SYSTEM_ERROR: <input type="checkbox"/> RETRY_WHEN_SYSTEM_ERROR: <input type="checkbox"/> SYSTEM_ERROR_CODES: RETRY_INTERVAL_IN_MINUTES: 0 RETRY_COUNT: 0
EMail	
Audit_IN	

Advanced Tab: Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

GEHS_CONFIG_FILEPATH	- Path of GEHS config file name –xml file containing parameters to be overridden at runtime.
STOP_THESE_FLOWS	<ul style="list-style-type: none"> - Leave blank to stop the current message flow. - Define comma separate flow name(s) to stop multiple flows. - Define Application%Flow if flow is running in Application/Library
STOP_FLOW_WHEN_ANY_ERROR	- Checked

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW		
Description		
Basic	EMAIL_ERROR <input checked="" type="checkbox"/>	
Advanced	EMAIL_SERVER_AND_PORT <input type="text" value="localhost:25"/>	
EMail	EMAIL_FROM <input type="text" value="wbiadm"/>	
Audit_IN	EMAIL_TO <input type="text" value="sudhakar.bodapati@ibm.com"/>	
	EMAIL_CC <input type="text"/>	
	EMAIL_BCC <input type="text"/>	
	EMAIL SUBJECT <input type="text"/>	
	EMAIL_ATTACHMENT_FILENAME <input type="text" value="/home/wbiadm/ftr/interface/interface.tsguide"/>	

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS CONFIG FILEPATH settings.

Expected Result:

If any error in the main flow logic the transaction will fail, and

- An email will be sent out to the contacts in EMAIL TO, EMAIL CC, EMAIL BCC attribute.



- The message will be put back into the input queue, and backout count of the message will be incremented by 1.
 - The flow will be stopped, Investigate the issue based on the email received, and once the issue is resolved start the message flow manually.
 - If the backout count on the message equals BOTHRESH defined on the queue, then the message will be routed the ERROR_QUEUE defined in the Basic Tab.
 - If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

3. Stop flow when a system error

Use this scenario to stop the message flow when a system error occurs in the message flow.

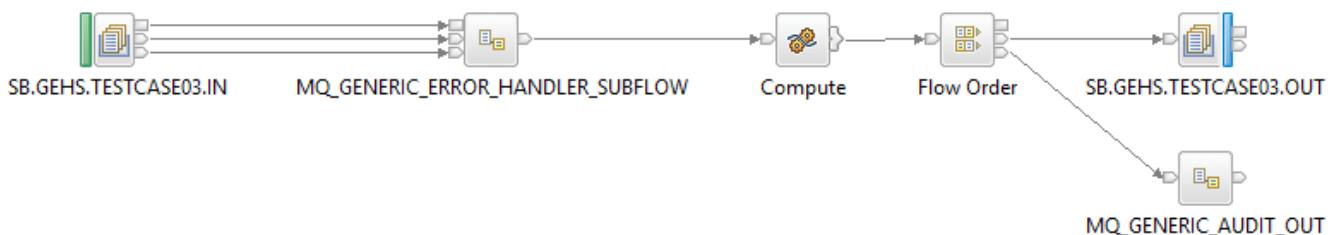
Typical system error codes are “database down”, “Queue not defined”, “Queue full”, “REST API / Web-Service is down”, etc.

Set BOTHRESH, and BOQNAME parameters of the Input Queue.

Set the BOQNAME to be same as the Input Queue name.

Example queue definition,

DEF QLOCAL(SB.GEHS.TESTCASE03.IN) BOQNAME(SB.GEHS.TESTCASE03.IN) BOTHRESH(10)



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	Queue name: SB.GEHS.TESTCASE03.ERROR Connection*: SERVER Destination queue manager name: <Integration node queue manager> Queue manager host name: Listener port number: Channel name: Security identity: Use SSL: SSL peer name: SSL cipher specification:
Advanced	
EMail	
Audit_IN	

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	GEHS_CONFIG_FILEPATH: /home/mesadmin/sudhakar/GEHS_TC3_STOP_FLOW_WHEN_SYSTEM_ERROR.xml
Advanced	STOP_THESE_FLOWS: <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> RETRY_WHEN_ANY_ERROR: <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> STOP_FLOW_WHEN_SYSTEM_ERROR: <input checked="" type="checkbox"/> <input type="checkbox"/> SYSTEM_ERROR_CODES: 2085,2053 RETRY_INTERVAL_IN_MINUTES: 0 RETRY_COUNT: 0
EMail	
Audit_IN	

Advanced Tab: Check STOP_FLOW_WHEN_SYSTEM_ERROR, and SYSTEM_ERROR_CODES

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	EMAIL_ERROR: <input checked="" type="checkbox"/>
Advanced	EMAIL_SERVER_AND_PORT: intrelay.ibm.com:25 EMAIL_FROM: wbiadm EMAIL_TO: sbodapati@esbtech.com EMAIL_CC: EMAIL_BCC: EMAIL_SUBJECT: ERROR - Tescase03 Message Flow EMAIL_ATTACHMENT_FILENAME: /home/wbiadm/ftr/interface/interface.tsguide
EMail	
Audit_IN	

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

Expected Result:

If any error in the main flow logic, then the transaction should fail, and

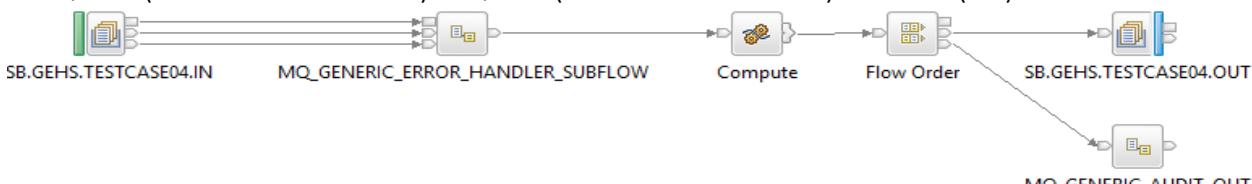
- An email will be sent out to the contacts in EMAIL_TO, EMAIL_CC, EMAIL_BCC.
- The message will be put back into the input queue, and the backout count of the message will be incremented by 1.
- The flow will be stopped if the exception matches the system error codes defined, if not matched then the message will be routed to the ERROR_QUEUE.
- Investigate the issue based on the email received, and once the issue is resolved manually start the message flow.
- When the backout count on the message equals BOTHRESH defined on the queue, then the message will be routed the ERROR_QUEUE.
- If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

4. Retry message when Any Error

Use this scenario to retry a failed message due to any error. The message will be retried based on the retry interval RETRY_INTERVAL_IN_SECONDS, and RETRY_COUNT. If the problem not resolved even after the number of RETRY_COUNT(s), the message will be routed to the ERROR_QUEUE.

Define the input queue as below;

```
DEF QLOCAL(SB.GEHS.TESTCASE04.IN) BOQNAME(SB.GEHS.TESTCASE04.IN) BOTHRESH(100)
```



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	Queue name: SB.GEHS.TESTCASE04.ERROR Connection*: SERVER Destination queue manager name: <Integration node queue manager> Queue manager host name: Listener port number: Channel name: Security identity: <input type="checkbox"/> Use SSL SSL peer name: SSL cipher specification:
Advanced	
EMail	
Audit_IN	

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	GEHS_CONFIG_FILEPATH: /home/mesadmin/sudhakar/GEHS_TC4_RETRY_WHEN_ANY_ERROR.xml
Advanced	STOP_THESE_FLOWS: <input type="checkbox"/> STOP_FLOW_WHEN_ANY_ERROR: <input checked="" type="checkbox"/> RETRY_WHEN_ANY_ERROR: <input checked="" type="checkbox"/> STOP_FLOW_WHEN_SYSTEM_ERROR: <input type="checkbox"/> RETRY_WHEN_SYSTEM_ERROR: <input type="checkbox"/> SYSTEM_ERROR_CODES: RETRY_INTERVAL_IN_MINUTES: 5 RETRY_COUNT: 10
EMail	
Audit_IN	

Advanced Tab: Parameters can be overridden by GEHS_CONFIG_FILEPATH file.

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW	
Description	
Basic	EMAIL_ERROR <input checked="" type="checkbox"/>
Advanced	EMAIL_SERVER_AND_PORT intrelay.ibm.com:25
EMail	EMAIL_FROM wbiadm
Audit_IN	EMAIL_TO sbodapati@esbtech.com
	EMAIL_CC
	EMAIL_BCC
	EMAIL SUBJECT ERROR - Tescase03 Message Flow
	EMAIL_ATTACHMENT_FILENAME /home/wbiadm/ftr/interface/interface.tsguide

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

Expected Result:

If any error in the main flow logic then the transaction will fail, and

- An email will be sent out to the contacts in EMAIL_TO, EMAIL_CC, EMAIL_BCC.
- The flow will be paused for RETRY_INTERVAL_IN_SECONDS seconds
- The message will be put back into the input queue, and the backout count of the message will be incremented by 1.
- The same message from the input queue will be retried by the flow again. If the retry is successful then the transaction will be completed, otherwise the message will be retried until RETRY_COUNT. If unsuccessful processing with in RETRY_COUNT then the message will be routed to the ERROR_QUEUE.
- When the backout threshold is reached (BOTHRESH) the message will be put into the ERROR_QUEUE.
- If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

5. Retry message when System Error

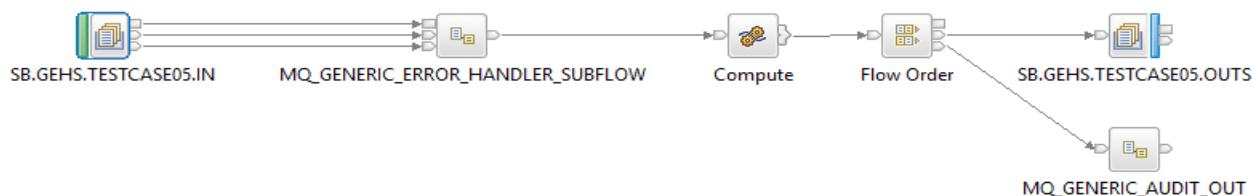
Use this scenario to retry a failed message due to a system error. The message will be retried based on the retry interval RETRY_INTERVAL_IN_SECONDS, and RETRY_COUNT. If the problem not resolved even after the number of RETRY_COUNT(s), the message will be routed to the ERROR_QUEUE.

Set BOTHRESH, and BOQNAME parameters of the Input Queue.

Set the BOQNAME to be same as the Input Queue name.

Example queue definition,

DEF QLOCAL(SB.GEHS.TESTCASE05.IN) BOQNAME(SB.GEHS.TESTCASE05.IN) BOTHRESH(100)



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description			More...	
Basic	Queue name	SB.GEHS.TESTCASE05.ERROR		Browse MQ Services...
Advanced	Connection*	SERVER		▼
EMail	Destination queue manager name	<Integration node queue manager>		
Audit_IN	Queue manager host name			
	Listener port number			
	Channel name			
	Security identity			
	Use SSL	<input type="checkbox"/>		
	SSL peer name			
	SSL cipher specification			

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description			More...
Basic	GEHS_CONFIG_FILEPATH	/home/mesadmin/sudhakar/GEHS_TC4_RETRY_WHEN_SYSTEM_ERROR.xml	
Advanced	STOP THESE FLOWS	<input type="text"/>	
EMail	STOP_FLOW_WHEN_ANY_ERROR	<input type="checkbox"/>	
Audit_IN	RETRY_WHEN_ANY_ERROR	<input checked="" type="checkbox"/>	
	STOP_FLOW_WHEN_SYSTEM_ERROR	<input type="checkbox"/>	
	RETRY_WHEN_SYSTEM_ERROR	<input type="checkbox"/>	
	SYSTEM_ERROR_CODES		
	RETRY_INTERVAL_IN_MINUTES	5	
	RETRY_COUNT	10	

Advanced Tab: Parameters can be overridden by GEHS_CONFIG_FILEPATH file.

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description			More...
Basic	EMAIL_ERROR	<input checked="" type="checkbox"/>	
Advanced	EMAIL_SERVER_AND_PORT	intrelay.ibm.com:25	
EMail	EMAIL_FROM	wbiadm	
Audit_IN	EMAIL_TO	sbodapati@esbtech.com	
	EMAIL_CC		
	EMAIL_BCC		
	EMAIL SUBJECT	ERROR - Tescase04 Message Flow	
	EMAIL_ATTACHMENT_FILENAME	/home/wbiadm/ftr/interface/interface.tsguide	

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

Expected Result:

If any error in the main flow logic then the transaction will fail, and

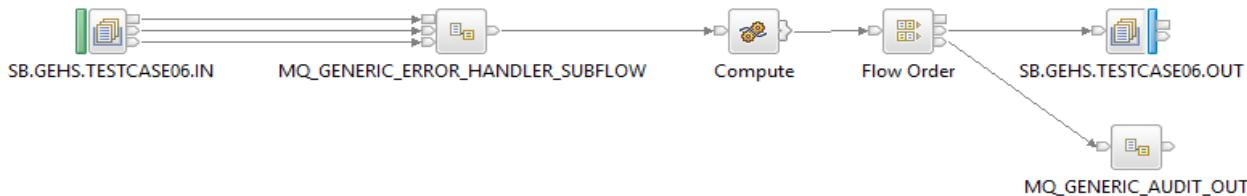
- An email will be sent out to the contacts in EMAIL_TO, EMAIL_CC, EMAIL_BCC.
- The flow will be paused for RETRY_INTERVAL_IN_SECONDS seconds
- The message will be put back into the input queue, and the backout count of the message will be incremented by 1.
- The same message from the input queue will be processed by the flow again, and will be retried. If successful processing the message then the transaction will be completed, otherwise the message will be retried until RETRY_COUNT. If unsuccessful processing within RETRY_COUNT then the message will be routed to the ERROR_QUEUE.
- When the backout count of the message equals BOTHRESH defined on the queue, then the message will be routed to the ERROR_QUEUE.
- If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

6. Retry message and Stop Flow when Any error

Use this scenario to retry, and stop the message flow. The failed message will be retried based on the RETRY_INTERVAL_IN_SECONDS, and RETRY_COUNT. The flow will be stopped after retrying "RETRY_COUNT" times.

Define the input queue as below;

```
DEF QLOCAL(SB.GEHS.TESTCASE06.IN) BOQNAME(SB.GEHS.TESTCASE06.IN) BOTHRESH(100)
```



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description	More...
Basic	Queue name: SB.GEHS.TESTCASE06.ERROR Browse MQ Services...
Advanced	Connection*: SERVER
EMail	Destination queue manager name: <Integration node queue manager>
Audit_IN	Queue manager host name Listener port number Channel name Security identity Use SSL SSL peer name SSL cipher specification

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description	More...
Basic	GEHS_CONFIG_FILEPATH: C:\workspace\gehs2\CommonSubFlows-2019\gehs_A_B_TESTCASE06.xml
Advanced	STOP_THESE_FLOWS: STOP_FLOW_WHEN_ANY_ERROR RETRY_WHEN_ANY_ERROR STOP_FLOW_WHEN_SYSTEM_ERROR RETRY_WHEN_SYSTEM_ERROR SYSTEM_ERROR_CODES RETRY_INTERVAL_IN_MINUTES: 5 RETRY_COUNT: 25
EMail	
Audit_IN	

Advanced Tab: Parameters can be overridden by GEHS_CONFIG_FILEPATH file.

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description	More...
Basic	EMAIL_ERROR: <input checked="" type="checkbox"/>
Advanced	EMAIL_SERVER_AND_PORT: intrelay.ibm.com:25
EMail	EMAIL_FROM: wbiadm
Audit_IN	EMAIL_TO: sbodapati@esbtech.com EMAIL_CC EMAIL_BCC EMAIL_SUBJECT: ERROR - Tescase03 Message Flow EMAIL_ATTACHMENT_FILENAME: /home/wbiadm/ftr/interface/interface.tsguide

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

Expected Result:

If any error in the main flow logic then the transaction will fail, and

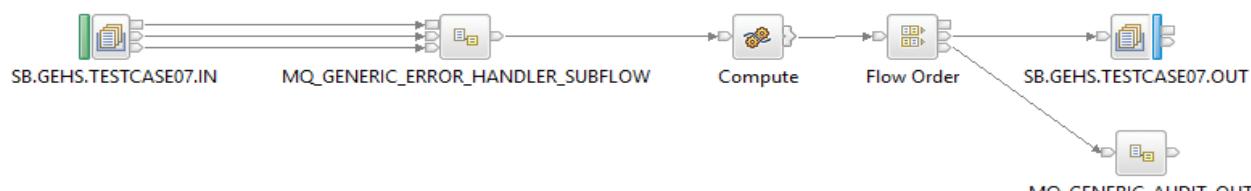
- An email will be sent out to the contacts in EMAIL_TO, EMAIL_CC, EMAIL_BCC.
- The flow will be paused for RETRY_INTERVAL_IN_SECONDS seconds
- The message will be put back into the input queue
- The message will be retried. If the retry is successful the transaction will be completed, otherwise the message will be retried until RETRY_COUNT. If unsuccessful within the RETRY_COUNT, then the flow will be stopped.
- When the backout threshold is reached (BOTHRESH) the message will be put into the ERROR_QUEUE.
- If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

7. Retry message and Stop the Flow when System error.

Use this scenario to retry a failed message due to a system type error, the message will be retried every RETRY_INTERVAL_IN_SECONDS. The message will be retried until the RETRY_COUNT. The flow will be stopped if the message couldn't be processed within the RETRY_COUNT.

Define the Input Queue as below;

```
DEF QLOCAL(SB.GEHS.TESTCASE07.IN) BOQNAME(SB.GEHS.TESTCASE07.IN) BOTHRESH(100)
```



Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description		
Basic	Queue name	SB.GEHS.TESTCASE07.ERROR <input type="button" value="Browse MQ Services..."/>
Advanced	Connection*	SERVER <input type="button" value="More..."/>
EMail	Destination queue manager name	<Integration node queue manager>
Audit_IN	Queue manager host name	
	Listener port number	
	Channel name	
	Security identity	
	Use SSL	<input type="checkbox"/>
	SSL peer name	
	SSL cipher specification	

Basic tab: Error Queue details above

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description		
Basic	GEHS_CONFIG_FILEPATH	C:\workspace\gehs2\CommonSubFlows-2019\gehs_A_B_TESTCASE07.xml
Advanced	STOP_THESE_FLOWS	<input type="text"/>
EMail	STOP_FLOW_WHEN_ANY_ERROR	<input type="checkbox"/>
Audit_IN	RETRY_WHEN_ANY_ERROR	<input type="checkbox"/>
	STOP_FLOW_WHEN_SYSTEM_ERROR	<input checked="" type="checkbox"/>
	RETRY_WHEN_SYSTEM_ERROR	<input checked="" type="checkbox"/>
	SYSTEM_ERROR_CODES	2085,2053
	RETRY_INTERVAL_IN_MINUTES	5
	RETRY_COUNT	25

Advanced Tab: Parameters can be overridden by GEHS_CONFIG_FILEPATH file.

Subflow MQ_GENERIC_ERROR_HANDLER_SUBFLOW Node Properties - MQ_GENERIC_ERROR_HANDLER_SUBFLOW

Description		More...
Basic	EMAIL_ERROR	<input checked="" type="checkbox"/>
Advanced	EMAIL_SERVER_AND_PORT	intrelay.ibm.com:25
EMail	EMAIL_FROM	wbiadm
Audit_IN	EMAIL_TO	sbodapati@esbtech.com
	EMAIL_CC	
	EMAIL_BCC	
	EMAIL SUBJECT	ERROR - Tescase03 Message Flow
	EMAIL_ATTACHMENT_FILENAME	/home/wbiadm/ftr/interface/interface.tsguide

Email Tab: Provide Email Attributes, Parameters can be overridden by GEHS_CONFIG_FILEPATH settings.

Expected Result:

If any error in the main flow logic, then the transaction will fail, and

- An email will be sent out to the contacts in EMAIL_TO, EMAIL_CC, EMAIL_BCC.
- The flow will be paused for RETRY_INTERVAL_IN_SECONDS seconds
- The message will be put back into the input queue
- The message from the input queue will be retried. If the retry is successful, then the transaction will be completed, otherwise the message will be retried until RETRY_COUNT. If unsuccessful within the RETRY_COUNT, then the flow will be stopped.
- When the backlog threshold is reached (BOTHRESH) the message will be put into the ERROR_QUEUE.
- If the ERROR_QUEUE is either full, or not available then the flow will be stopped, and the message will be put back into the Input Queue.

Acknowledgements

The author would like to thank Mike (Shenfu) Fan from the IBM Software Services for WebSphere (ISSW) team, and Forest (Mu) Lin from the IBM Watson IOT team for reviewing and helping to improve this article.

Downloadable resources

Description	Name	Size
Project Interchange	IIB_MQInput_GEHS_V1_0.zip	96kb
pdf document	IIB_MQInput_GEHS_v1_0.pdf	701kb