ERROR HANDLING USING RETRY MECHANISM GUIDE

1. INTRODUCTION

As part of Error handling in ThreadConnect, the failed flowfiles are either re-processed or logged and reported. Basic re-processing of flowfile is achieved by routing back the failure relationship of processor back to itself, thereby allowing failed flowfiles to be retried by the processor. This approach gives users less control over the number of retry attempts. The main aim of the Retry flow developed is to facilitate users to customize and control over the re-processing of failed flowfiles.

2. RETRY FLOW DESCRIPTION

Retry flow designed to support (as of now) two types of failures namely Technical (like when target system like ERP, Database is down, which results in data not being accessible nor to process, API call returns 500 as response code due to internal error as the API was not able to process the request etc.) and Functional failures (Sometimes data ingestion may fail due to the ID corresponding to the data not being added in the target system).

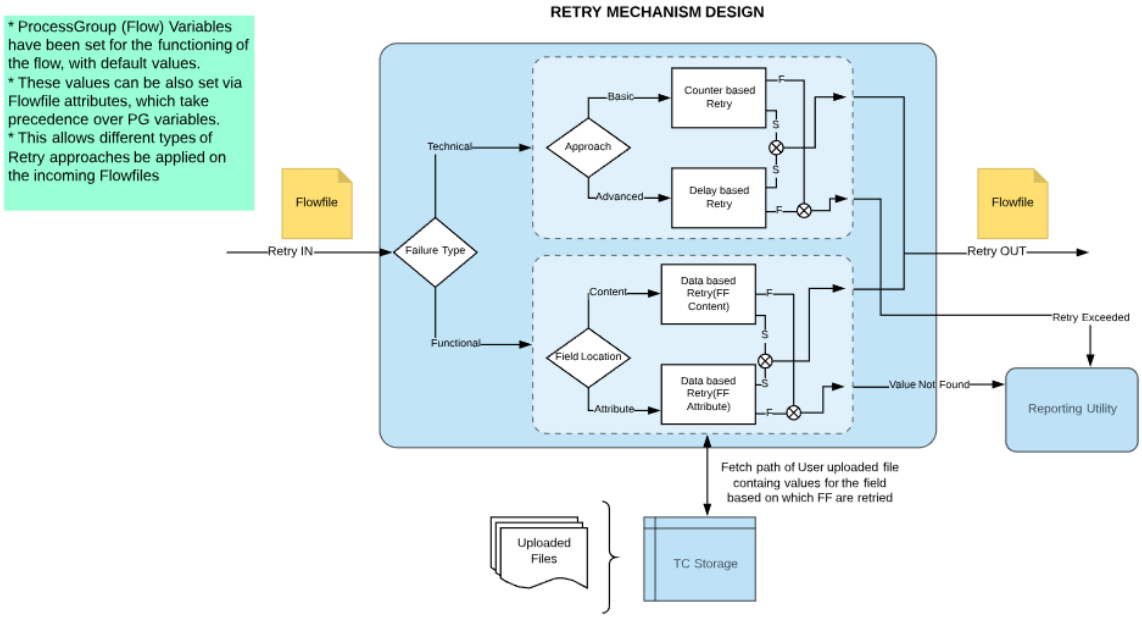

Figure: Design for ThreadConnect' s Retry Flow

Error Handling Types

1. Technical Failures –

    a) *Retry with Count*:- Basic Error Handling approach where flowfiles are retried based on Retry.Counter value Initialized. If the Retry fails after specified number of times, based on the type of reporting needed either an email can be triggered to the configured email id/ distribution list with configured, subject and body or

ServiceNow Incident is generated based upon configured properties of Urgency, Configuration Item, Short Description, and Long Description and User SSO along with an email.

b) *Retry with Delay*:- Advanced retry based Error Handling approach where flowfiles are retried based on Retry.Counter value Initialized. There is a progressive delay between each retry. Ex. The first retry is delay for 1m, second retry is delayed by 15m and so on based on number of retries. If the Retry fails after specified number of times, based on the type of reporting needed either an email can be triggered to the configured email id/ distribution list with configured, subject and body or ServiceNow Incident is generated based upon configured properties of Urgency, Configuration Item, Short Description, and Long Description and User SSO along with an email.

2. Functional Failures – Advanced Error Handling approach where flowfiles are retried based on a Field value. List of accepted values for the field is provided by the user in a file which is used as lookup to check if the failed flowfile has the field value (present either in the flowfile content or as flowfile attribute) equal to any of the values mentioned in the list. If value is present, the flowfile is re-processed else its either logged or reported (either an email can be triggered to the configured email id/ distribution list with configured, subject and body or ServiceNow Incident is generated based upon configured properties of Urgency, Configuration Item, Short Description, and Long Description and User SSO along with an email) based on user's requirement.

Retry Flow Modules

- **Retry Mechanism Template** – This Root Process Group operates on Retry.Type variable based on which failed flowfiles are directed according to the type of failure (either Technical or Functional failure). This Process Group requires the following variables set by the user.

Variable Name	Variable Description
Retry.Type	Specifies the type of the Retry mechanism which has to be applied for the incoming failed flowfiles. (FunctionalFailure, TechnicalFailure)
TC.Environment	Specifies the ThreadConnect Environment where the Retry Mechanism flow is running. (ThreadConnect DEV/STAGE/QA/PROD)

- **Retry Technical Failure** – This Process Group operates on Retry.Technical.Approach variable based on which the type of Error Handling approach should be applied for the failed flowfiles due to Technical failures. This Process Group requires the following variables set by the user.

Variable Name	Variable Description
Retry.Technical.Approach	Specifies the approach of the Technical Failures which has to be applied for the incoming failed flowfiles. (RetryWithCount, RetryWithDelay)
Retry.Maxcount	Specifies the maximum number of retry attempts for the incoming failed flowfiles.
Email.From	This is for Email reporting. Mentions the ID from which email is being sent.
Email.To	This is for Email reporting. Mentions the ID to which email has to be sent.
Error.Code	This is for Email reporting. Specifies the Error code for the type of error.
Error.Description	This is for Email reporting. Specifies details about the Error.
Raise.ServiceNow.Ticket	This is for reporting. Specifies whether ServiceNow incident should be raised or not. (Yes, No)

a) **Retry With Count (Technical Failures)** – This Process Group operates on Retry.Counter and Retry.Maxcount variables. Whenever the Retry.Counter is less than or equal to the Retry.Maxcount configured, it retries the flowfile by routing it back to the processor (source of error). Once Retry.Counter exceeds the Retry.Maxcount configured, it then routes it to a failure relation and a. either EmailUtility is invoked or ServiceNowIncidentUtility and EmailUtility is invoked based on Raise.ServiceNow.Ticket variable, b. The flowfile is kept in memory for 10 hours, the Ops Team should decide whether to re-process it manually beyond the configurable retries (say once the source/ target is up etc). For which

AdminToEnable_ForcedProcessing_UpdateAttribute processor should be enabled and started and manual reprocessing will happen automatically.

b) **Retry With Delay (Technical Failures)** – This Process Group operates on Retry.Counter Retry.Maxcount and Sleep.Time variables. Whenever the Retry.Counter is less than or equal to the Retry.Maxcount configured, it retries the flowfile by routing it back to the processor (source of error) after waiting for certain amount of time based on the Sleep.Time variable. Once Retry.Counter exceeds the Retry.Maxcount configured, it then routes it to a failure relation and a. either EmailUtility is invoked or ServiceNowIncidentUtility and EmailUtility is invoked based on Raise.ServiceNow.Ticket variable, b. The flowfile is kept in memory for 10 hours, the Ops Team should decide whether to re-process it manually beyond the configurable retries (say once the source/ target is up etc). For which AdminToEnable_ForcedProcessing_UpdateAttribute processor should be enabled and started and manual reprocessing will happen automatically.

c) **Email Utility** – Sends an automated email to the directed recipient with the details of the error. Variables such as from, to, integration name, error message, error description etc. are all dynamically initialized.

d) **ServiceNowIncidentUtility** – It is an automated Incident creation utility. Variables like short desc, long desc, work notes, integration name, service, environment are dynamically initialized basis user's configuration. ServiceNow API exposed via akana is invoked. Incident number and assignment group are returned in API response and sent to email utility as attributes to notify the end user if Raise.ServiceNow.Ticket variable is set to 'Yes'. This Process Group requires the following variables set by the user.

Variable Name	Variable Description
SNow.Desc	Used to populate the description in ServiceNow tool.
SNow.Short.Desc	Used to populate the short description in ServiceNow tool.
SNow.Work.Notes	Used to populate the work notes in ServiceNow tool.
SNow.Environment	Used to populate the environment for which incident is being reported. Allowed values ThreadConnect-dev,

	ThreadConnect-stage, ThreadConnect-qa, ThreadConnect-prod.
SNow.Service	Used to populate the service for which incident is being reported allowed values ThreadConnect.
SNow.SSO	Used to populate the SSO of the individual creating the incident.
SNow.Urgency	Used to populate the urgency for the incident. Allowed values 1,2,3 for P1,P2,P3 incidents respectively.

- **Retry Functional Failure** – This Process Group operates on Retry.Functional.Approach variable, specifies where field is in the failed flowfile (either in flowfile content or as flowfile attribute) based on which the check for the field's value is done with the list of approved values for the field present in the file provided by the user. If none of the values match the field value of the flowfile then it's routed to failure relationship and a. either the flowfile is logged or reported (EmailUtility is invoked or ServiceNowIncidentUtility and EmailUtility is invoked based on Raise.ServiceNow.Ticket variable), b. The flowfile is kept in memory for 10 hours, the Ops Team should decide whether to re-process it manually (say once the source/ target is being added with the missing values for the field etc). For which AdminToEnable_ForcedProcessing_UpdateAttribute processor should be enabled and started and manual reprocessing will happen automatically. This Process Group requires the following variables set by the user.

Variable Name	Variable Description
Retry.Functional.Approach	Specifies the location of field based on which flowfile is
Retry.Field.Name	Specifies the name of the field whose value has to be looked up within the FF attribute or content to perform retry. (Should be the same as mentioned in attribute or in the content of FF).
Retry.LookupFile.Name	Specifies the file name containing the list of values for the field, that have to be retried. Ex. sample.txt

Report.Failure	This is for reporting. Specifies whether or not reporting has to be done for flowfiles that don't match the list of values for the field. (Yes, No)
Raise.ServiceNow.Ticket	This is for reporting. Specifies whether ServiceNow incident should be raised or not. (Yes, No)

a) **Email Utility** – Sends an automated email to the directed recipient with the details of the error if Report.Failure variable is set to 'Yes'. Variables such as from, to, integration name, error message, error description etc. are all dynamically initialized. This Process Group requires the following variables set by the user.

Variable Name	Variable Description
Email.From	This is for Email reporting. Mentions the ID from which email is being sent.
Email.To	This is for Email reporting. Mentions the ID to which email has to be sent.
Error.Code	This is for Email reporting. Specifies the Error code for the type of error.
Error.Description	This is for Email reporting. Specifies details about the Error.
Raise.ServiceNow.Ticket	This is for reporting. Specifies whether ServiceNow incident should be raised or not. (Yes, No)

b) **ServiceNowIncidentUtility** – It is an automated Incident creation utility. Variables like short desc, long desc, work notes, integration name, service, environment are dynamically initialized basis user's configuration. ServiceNow API exposed via akana is invoked. Incident number and assignment group are returned in API response and sent to email utility as attributes to notify the end user if both Report.Failure variable and Raise.ServiceNow.Ticket variable is set to 'Yes'. This Process Group requires the following variables set by the user.

Variable Name	Variable Description
SNow.Desc	Used to populate the description in ServiceNow tool.
SNow.Short.Desc	Used to populate the short description in ServiceNow tool.
SNow.Work.Notes	Used to populate the work notes in ServiceNow tool.
SNow.Environment	Used to populate the environment for which incident is being reported. Allowed values ThreadConnect-dev, ThreadConnect-stage, ThreadConnect-qa, ThreadConnect-prod.
SNow.Service	Used to populate the service for which incident is being reported allowed values ThreadConnect.
SNow.SSO	Used to populate the SSO of the individual creating the incident.
SNow.Urgency	Used to populate the urgency for the incident. Allowed values 1,2,3 for P1,P2,P3 incidents respectively.

3. TEMPLATE REQUIREMENTS

1. Drag the template icon, select the Retry_Mechanism_Template from the drop-down option. This would add the flow onto the canvas.

2. Retry flow requires Integration.Name value, which should be provided as flowfile attribute using UpdateAttribute processor (whose configuration is shown in the below image) , as a step in their flow before sending flowfiles to the retry flow by the flow from which the flowfile is coming from.

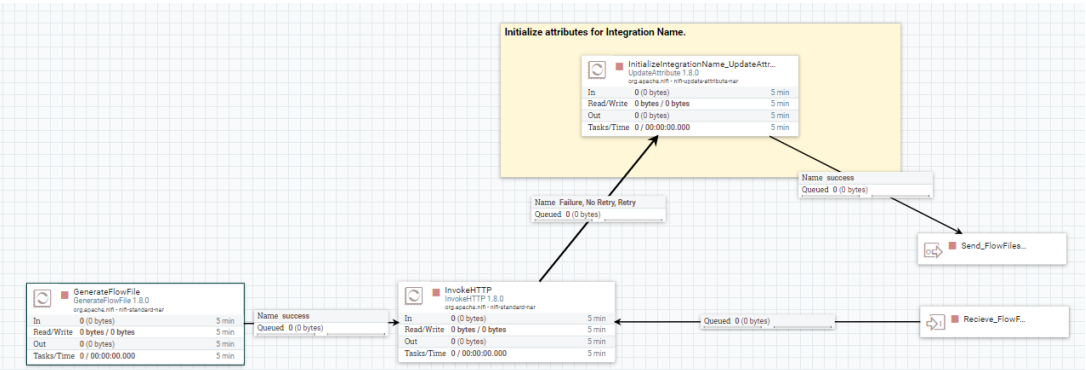

Figure: Add Integration.Name as Flowfile attribute within Source flow

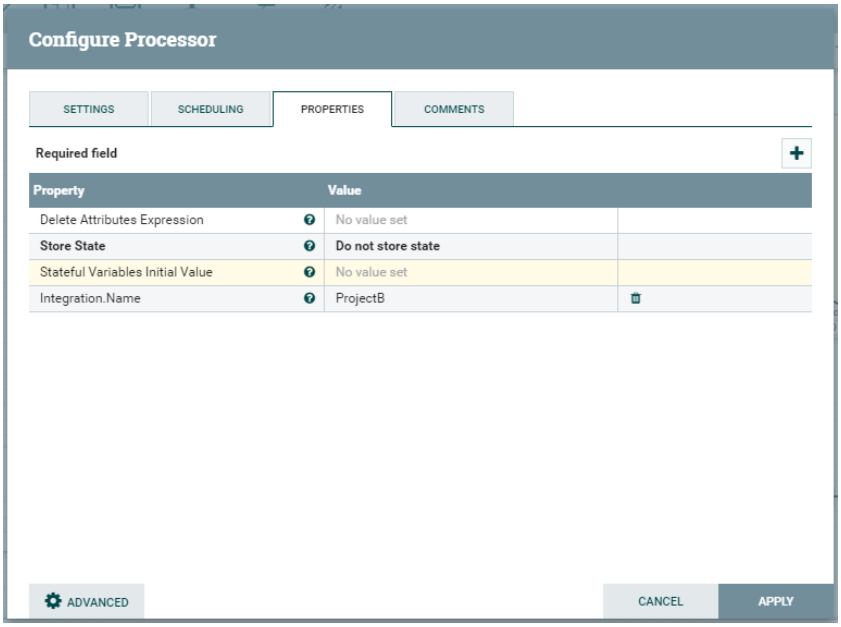

Figure: UpdateAttribute processor's configuration for setting Integration.Name property

3. Retry flow can accept multiple inputs from different Integration flows. Users can either use the default values for Process Group variables if the failed flowfiles coming from different integration flows or users can include UpdateAttribute processor, as a step in their flow before sending flowfiles to the retry flow to provide values for these variables as **flowfile attributes which takes precedence over the default variable values set for Process Group**. Hence, retry flow has the capability to support multiple inputs from different sources.

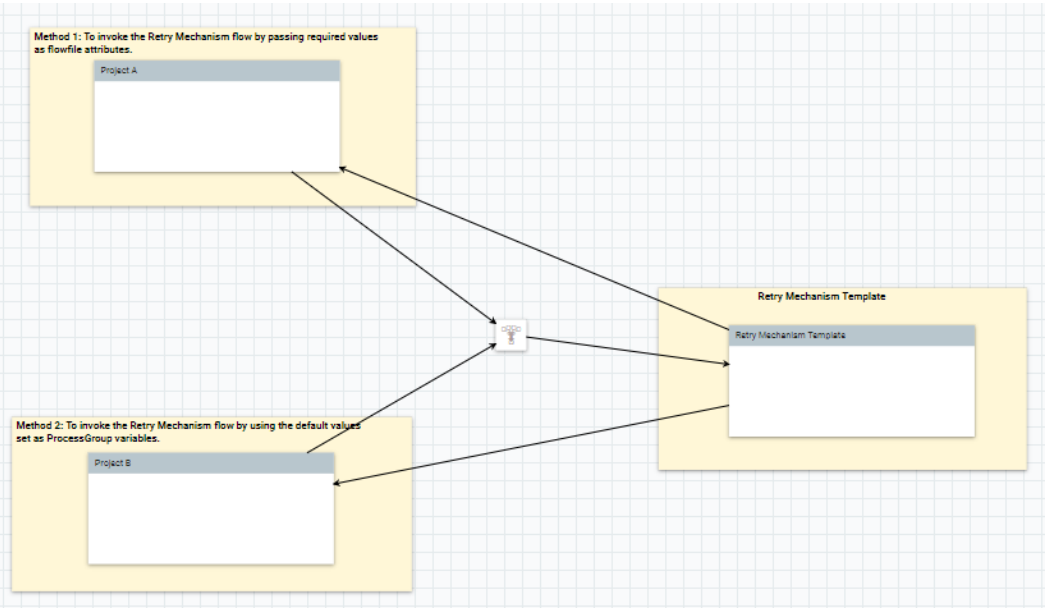

Figure: Multiple Input support for Retry flow

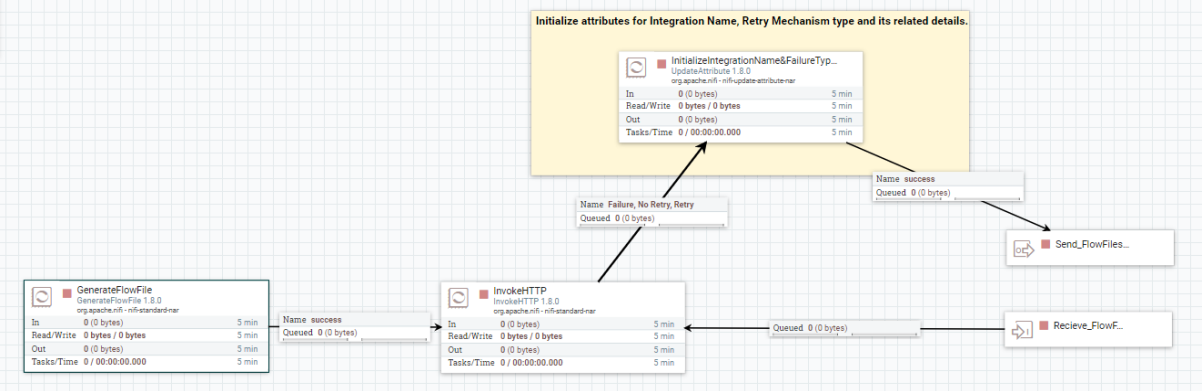

Figure: Process Group variable's value is provided as Flowfile attributes within Source Flow

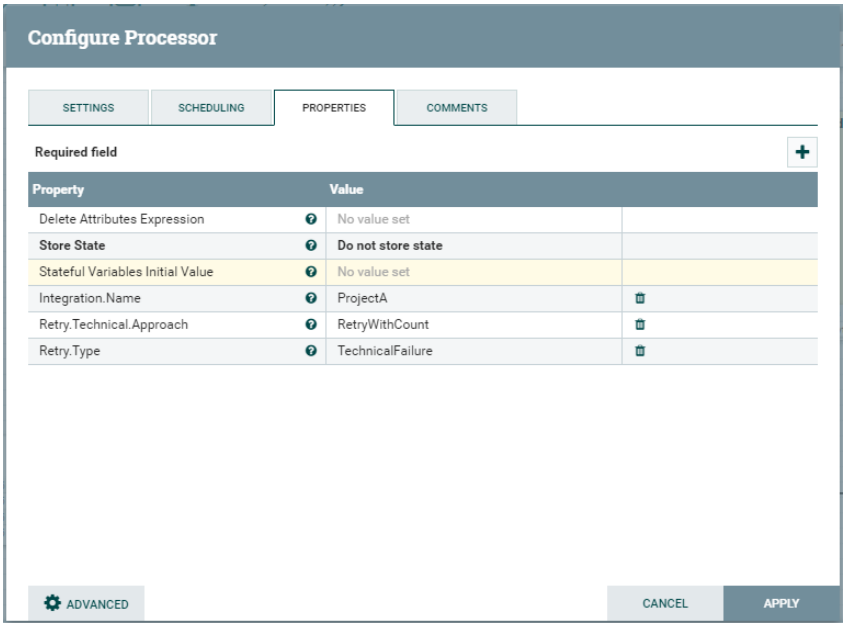

Figure: UpdateAttribute processor's configuration for the above scenario

4. In case of multiple source inputs from different integration flows, users should include a condition for each Integration flow input in the RouteOnAttribute processor within the Retry Mechanism Template Process Group. This is required as to create a separate connection to allow the processed flowfiles to route back to their source flow based on their Integration Name using the Integration.Name value.

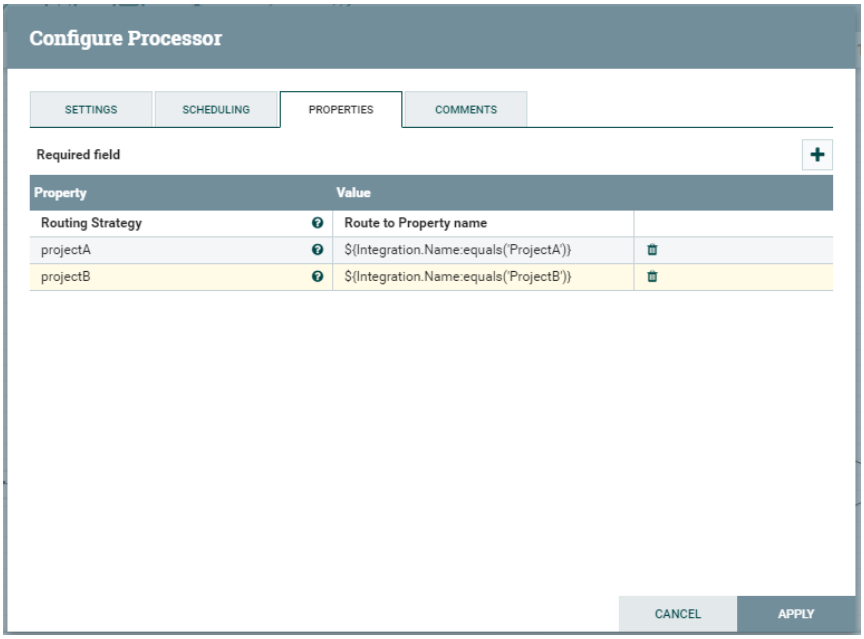

Figure: RouteOnAttribute processor's configuration for the above scenario

5. Based on the environment in which Retry flow is running, provide necessary value for the TC.Environment variable of Retry Mechanism Template Process Group.

6. In case of Functional failure user must provide the file name as the value for Retry.LookupFile.Name variable. The file is uploaded via 'Upload File' option. The file must contain each value separated by a new line for the processing of these values.

   Ex. Sample.txt
   yutyu-123
   oiays-345
   ersyis-234

   In case of local ThreadConnect instance Retry.LookupFile.Name should be provided with the absolute path of the file (present in the local storage of the system) as value, which requires minor changes to be made by the user in the groovy script in both ScriptToCheckIfFieldPresentInFFContent_ExecuteScript and ScriptToCheckIfFieldPresentInFFAttribute_ExecuteScript processors. Comment out the following lines:
   def expression = "\${UserFileDirectory('" + fileName_value + "')}"
   def temp_value = context.newPropertyValue(expression)
   def filePath_value = temp_value.evaluateAttributeExpressions().value

And replace temp_value with fileName_value
def filePath_value = temp_value.evaluateAttributeExpressions().value to def filePath_value = fileName_value.evaluateAttributeExpressions().value
The above changes have to be made to support local ThreadConnect instance.

In other cases, Retry.LookupFile.Name should have file name (which has been uploaded via 'Upload File') whose absolute path is determined internally using expression language function ${UserFileDirectory(filename)} .

7. Users should provide necessary values for the following processors SendEmail_InvokeHTTP (Email API), GetAkanaToken_InvokeHTTP (Akana API), SNowCreateIncident_InvokeHTTP (ServiceNow API) for the EmailUtility and ServiceNowIncidentUtility Process Groups to work.

8. To run the Retry flow, Right click on the Retry Mechanism Template Process Group, and select 'Start'.

9. The Retry flow can be customized based on user requirements.