

UiPath REFramework

UiPath REFramework Adaption to In-Process Queue

Sample implementation:

<https://github.com/rpapub/PurposefulPromethium/tree/windows/refactoring/v4>

InjectInProcessQueue.xaml

The screenshot displays the Visual Studio IDE with the XAML workflow editor open for `InjectInProcessQueue.xaml`. The workflow consists of the following steps:

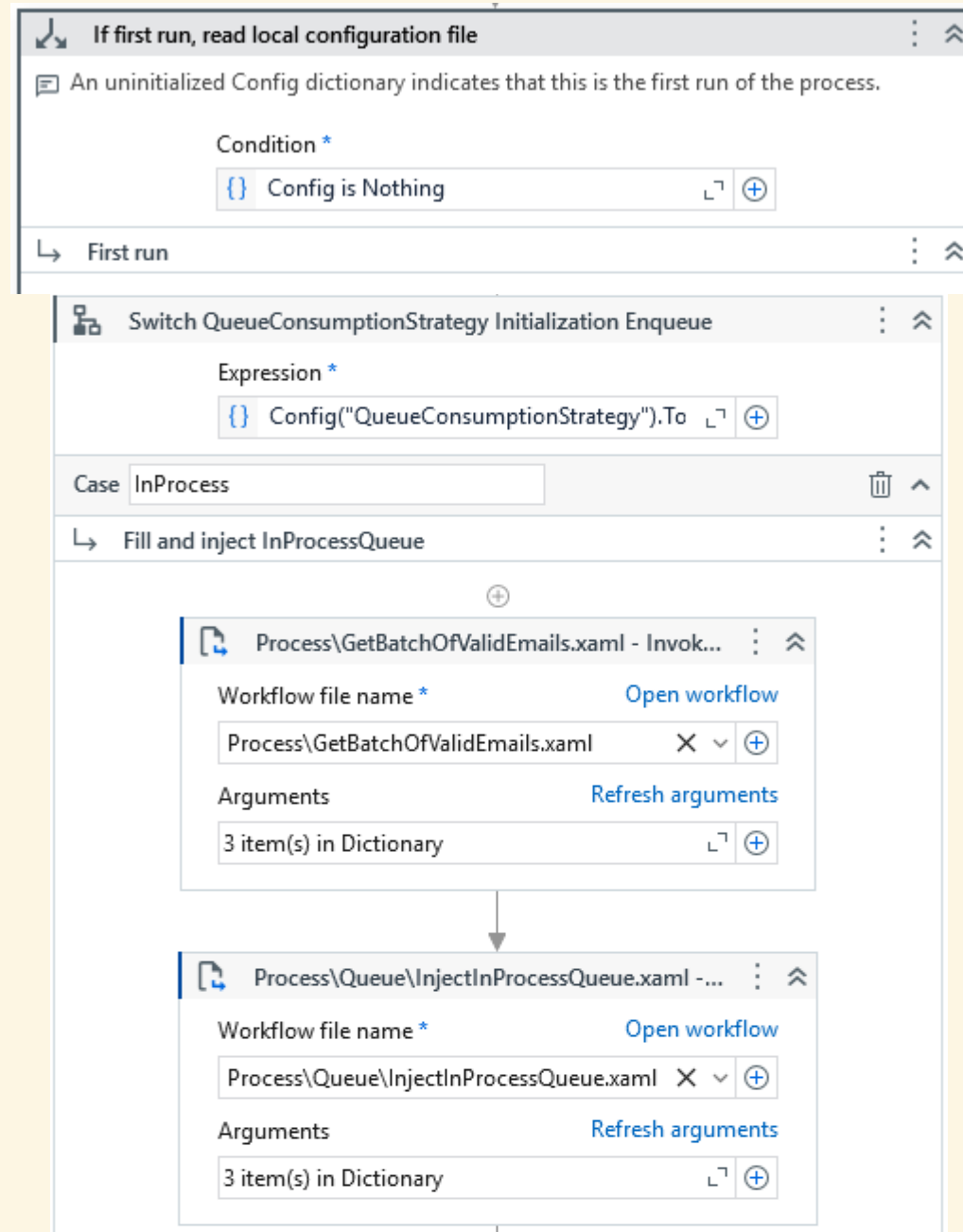
- Log Message INFO begin**: A message box with the text "Going to build and fill a in-process queue" and a log level of "Info".
- Assign out_InProcessQueue**: An assignment statement where the value `New Queue(Of QueueItem)` is assigned to the variable `out_InProcessQueue`.
- For Each mail in MailmessagesIncoming**: A loop that processes each mail. The body of the loop contains a comment: "each mail, if it has a Message-ID, that is used as QueueItem.Reference and added to the in-process queue." and a variable `in_MailmessagesIncoming`.

The Properties window on the right shows the configuration for the `System.Activities.Statements.Assign` statement:

- Common**: `DisplayName` is `Assign out_InProcessQueue`.
- Misc**: `Private` is unchecked. `Set value` is `New Queue(Of QueueItem)`. `To variable` is `out_InProcessQueue`.

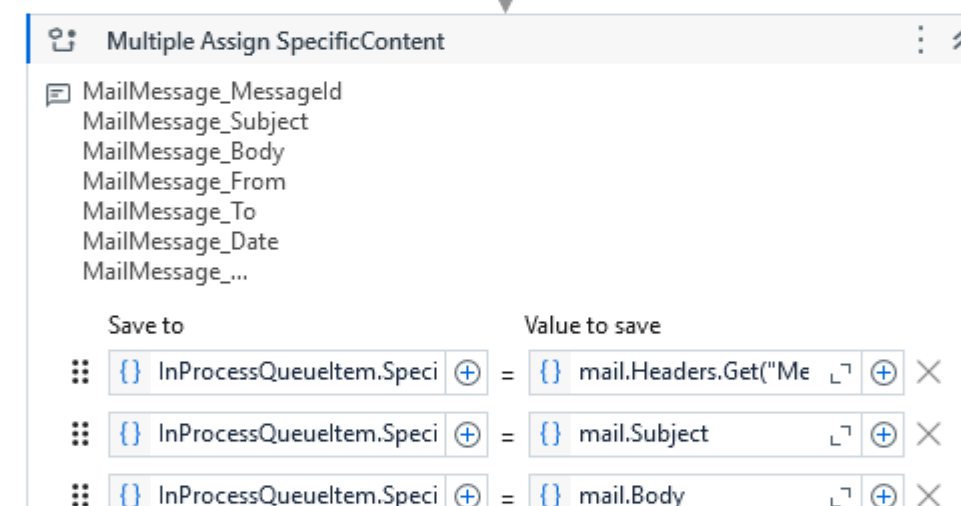
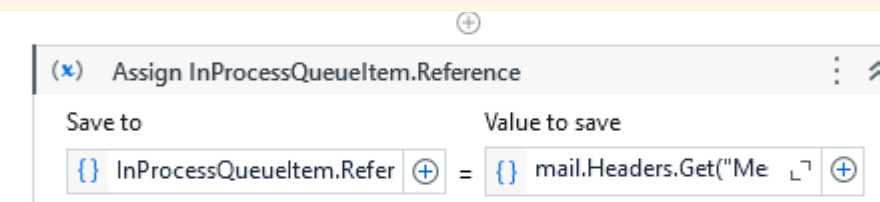
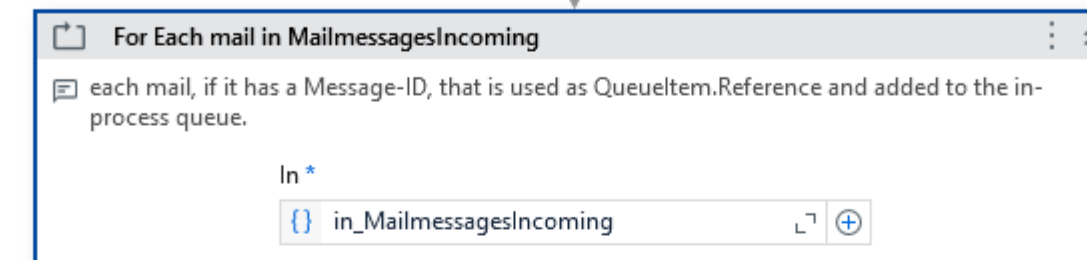
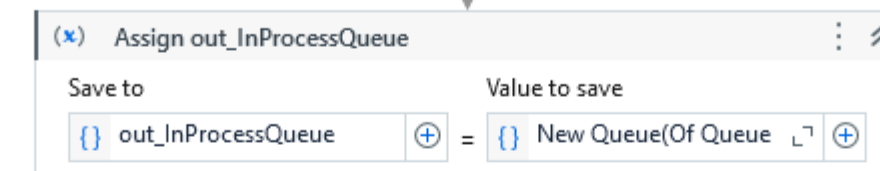
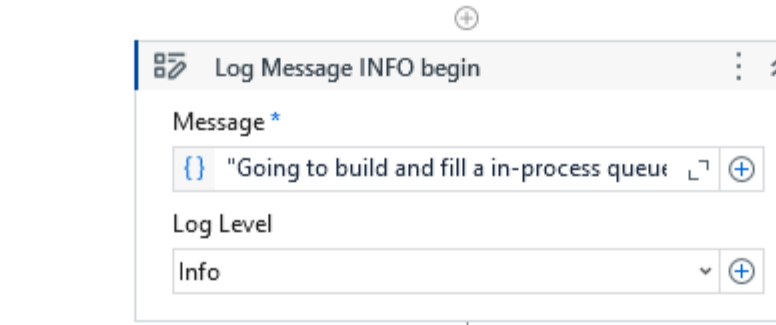
The bottom status bar shows the project name `PurposefulPromethium` and the file `refactoring/v5-migrate-win...`.

On first run, setup a in-process queue

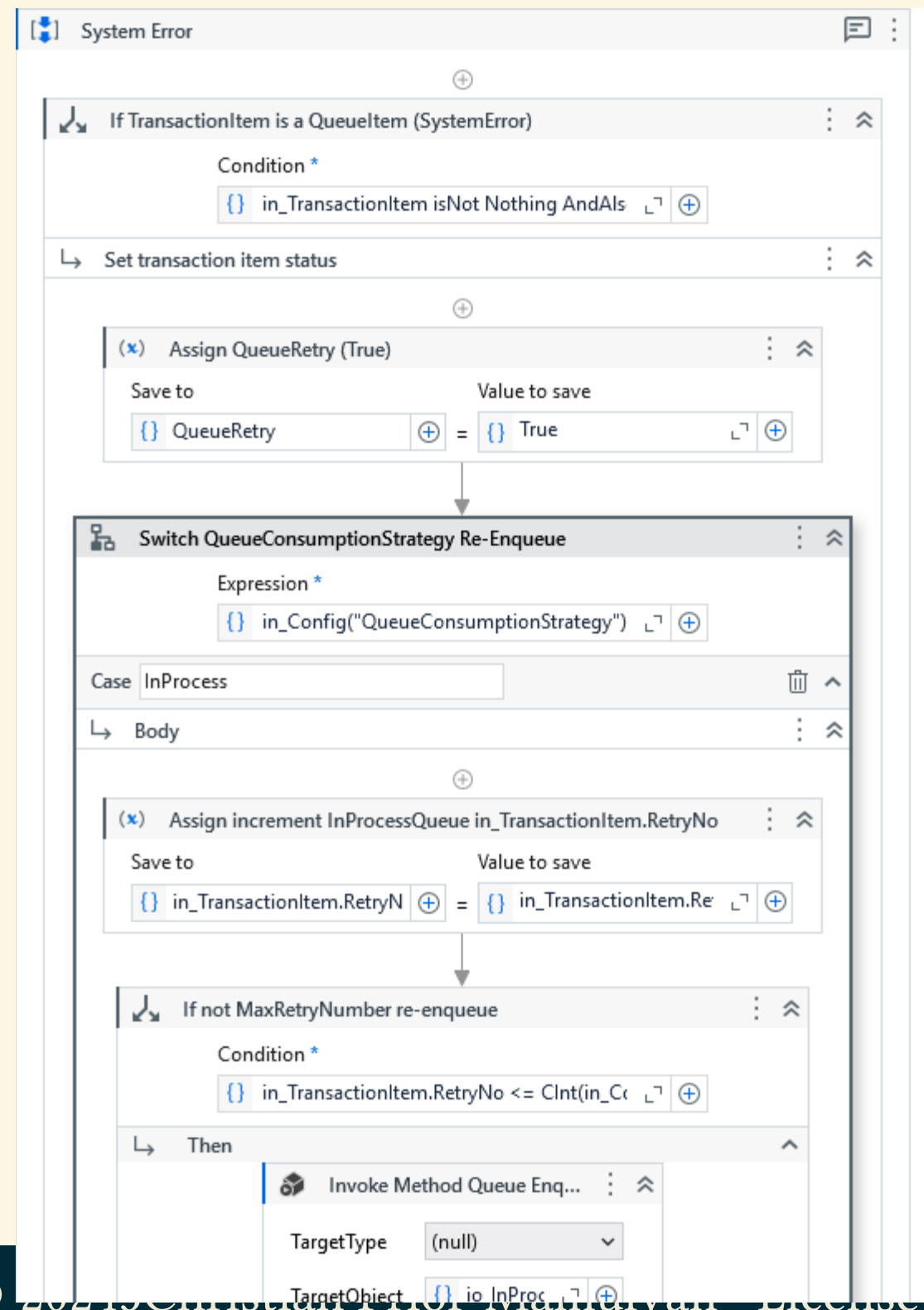


Designed to initialize and activate an alternative, in-process queue to manage items directly within the automation runtime, bypassing the default Orchestrator-managed queues. Reasons:

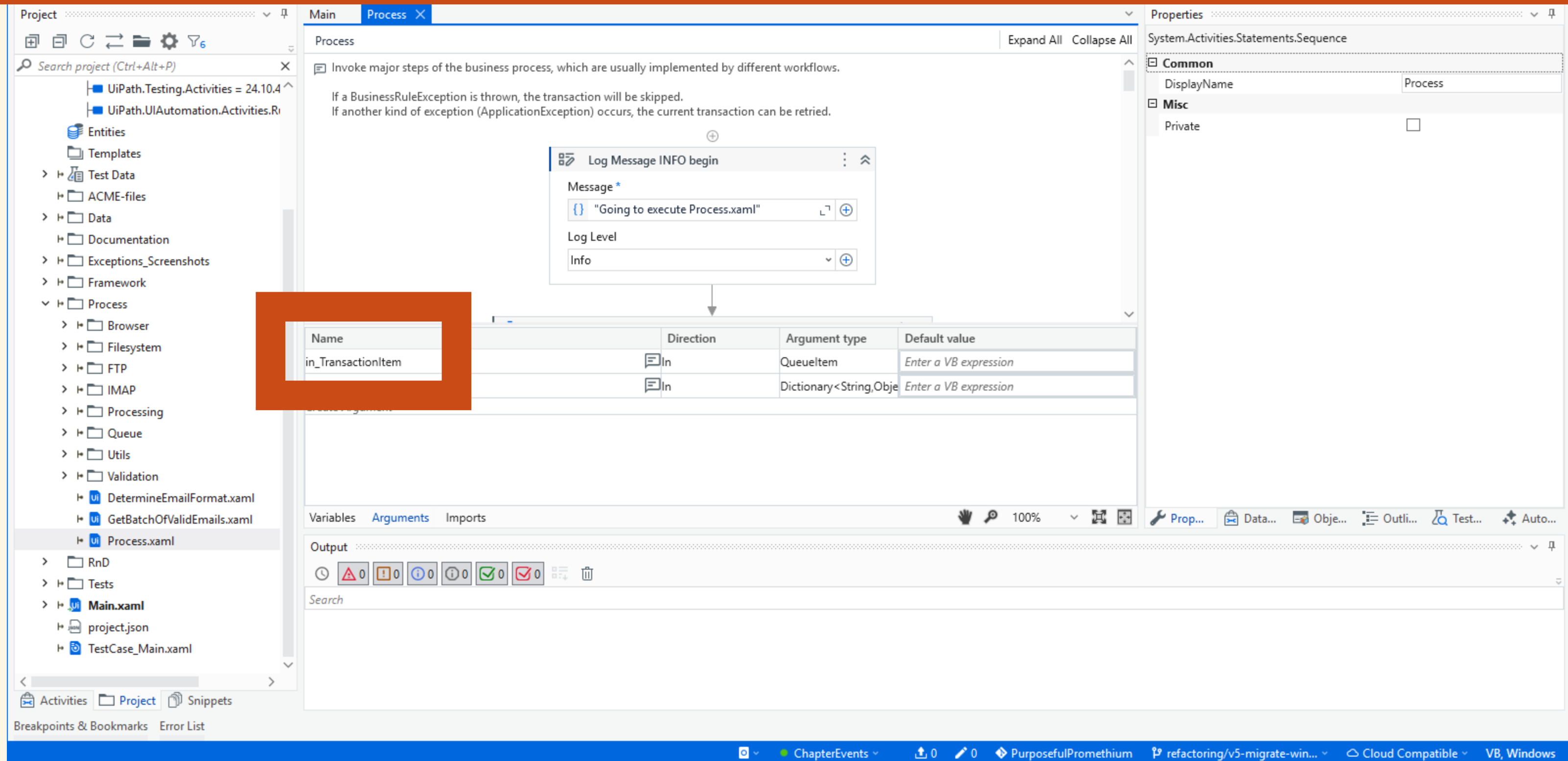
- Data Volatility: Emails in a mailbox can change frequently due to real-time human interaction
- Error Handling and State Management: In mailbox automations, the state of each MailMessage (e.g., read/unread, archived) affects processing



Key Aspect: Retries can work as expected



Process.xaml doesn't even notice the change



The screenshot displays the UiPath Studio interface with the **Process.xaml** workflow loaded. The **Log Message INFO begin** activity is highlighted with a red box, and its **Message** property is set to `"Going to execute Process.xaml"`. The **Arguments** tab is selected, showing a table with two arguments:

Name	Direction	Argument type	Default value
in_TransactionItem	In	QueueItem	Enter a VB expression
in_TransactionItem	In	Dictionary<String, Object>	Enter a VB expression

The **Properties** pane on the right shows the **Common** and **Misc** sections. The **Output** pane at the bottom shows a search bar and a list of output items.