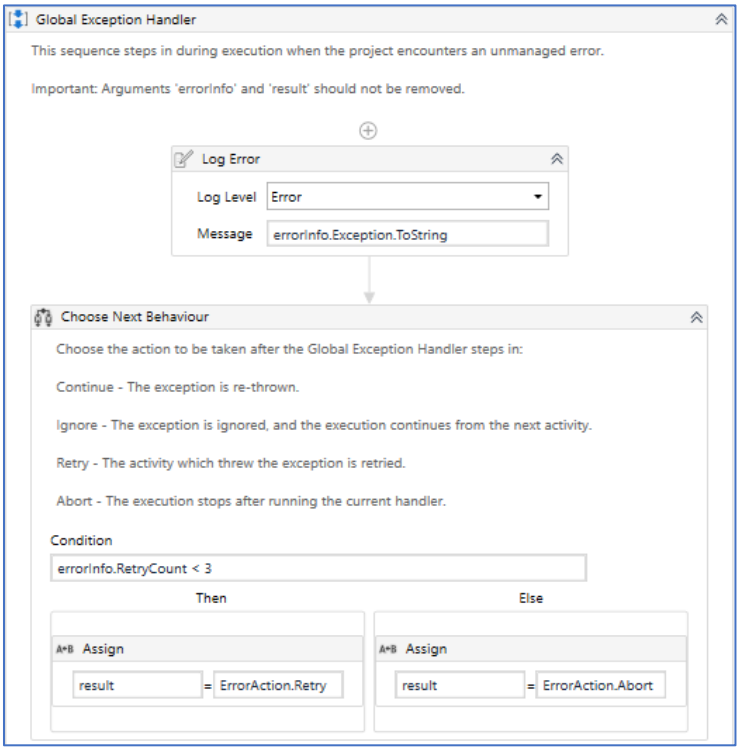
RPA Specialization

Course 4: Control Flow in RPA

Global Exception Handler

Global Exception Handler

The Global Exception Handler is a workflow designed to determine the behavior when encountering an execution error at the project level. Therefore, only one Global Exception Handler can be set per automation project.



The Global Exception Handler is used in conjunction with Try Catch. Only uncaught exceptions will reach the Exception Handler. If an exception occurs inside a Try-Catch activity and is successfully caught and treated inside the Catch block (and not re-thrown), it will not reach the Global Exception Handler.

Please note that it is available only for processes and not for library projects.

A Global Exception Handler can be created either by starting a new project or by setting an existing project as Global Exception Handler from the Project panel.

The Global Exception Handler has a predefined structure.

It has two **predefined arguments** (that shouldn't be removed):

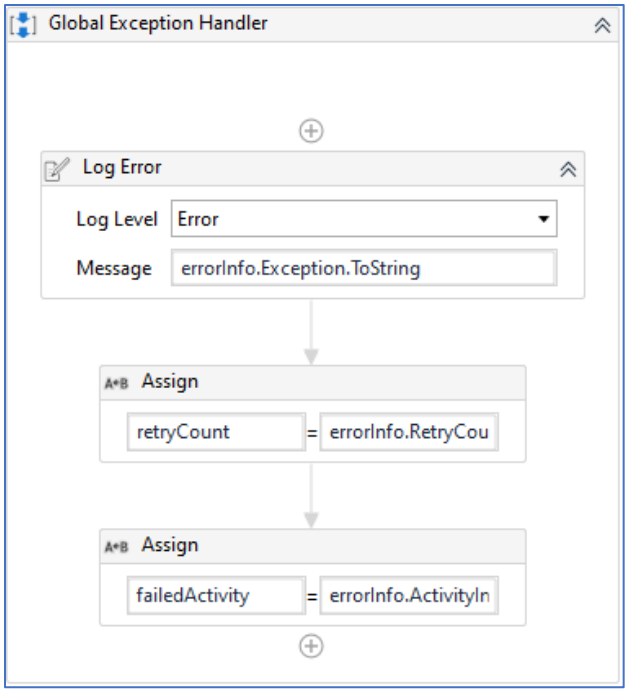
- errorInfo, with the In direction - contains the error that was thrown and the workflow that failed
- result, with the Out direction - will store the next behavior of the process when it encounters the error

It contains two **predefined actions** (that can be removed, and other actions can be added):

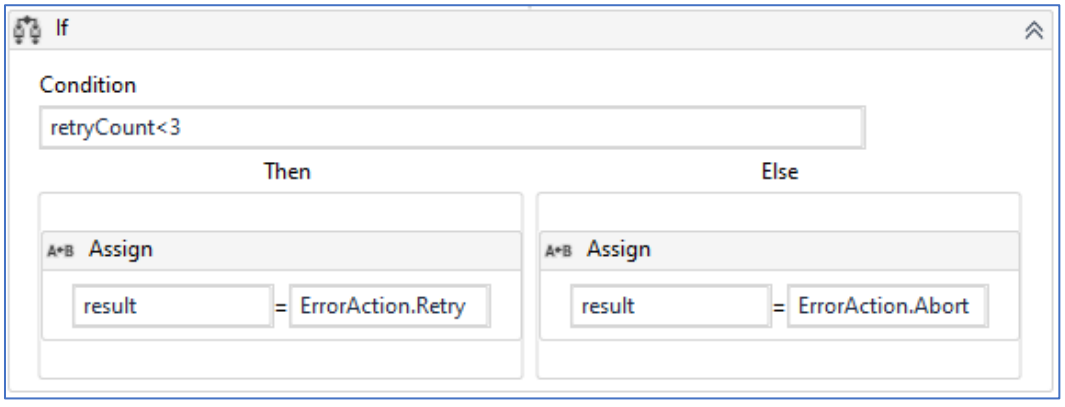
- Log Error: This part logs the error. The developer gets to choose the logging level: Fatal, Error, Warning, Info, and so on
- Choose Next Behavior: Here, the developer can choose the action to be taken when an error is encountered during execution:
    - Continue - The exception is re-thrown
    - Ignore - The exception is ignored, and the execution continues from the next activity
    - Retry - The activity which threw the exception is retried
    - Abort - The execution stops after running the current handler

Let's now take a look at how we can add a Global Exception handler to our process.

1. In the Design tab part of the Ribbon, select New > Global Handler. The New Global Handler window opens. Please type in a Name for the handler and save it in the project path. Click, Create a Global Exception Handler is added to the automation project.
2. In the Variables tab, create an Int32 variable named retryCount, set its scope to the entire Global Exception Handler, and leave the Default value field empty. The variable stores the number of retries, increasing the number by 1 each time an error is encountered.
3. Create a String variable named failedActivity and set its scope to the entire workflow. This variables stores the name of the activity which faulted and reveals it in the Output panel.
4. From the Activities panel, add an Assign activity below the Log Error in the Global Handler sequence. Add the retryCount to the activity and assign it the errorInfo.RetryCount expression, which counts the number of times the exception is thrown.
5. Add another Assign activity in which assign the errorInfo.ActivityInfo.Name expression to the failedActivity variable. At this point, the workflow should look like this:



6. Add an If activity to the workflow. Set the Condition to retryCount < 3 to limit the number of retries to 3.
7. In the Assign activity part of the Then category, assign the ErrorAction.Retry to the result variable to have the Global Exception Handler retry the activity which gave the exception.
8. In the Else category, assign the ErrorAction.Abort the result to have the Global Exception Handler abort the execution when the condition is met. The If activity should look like this:



When the **Global Exception Handler** encounters an exception, it logs the name of the activity which created the exception and starts retrying the activity three times. If it encounters the same

exception each time and the number of retries reaches 3, the execution is aborted at the activity level that threw the exception.

If an exception isn't encountered during one of the retries, the workflow's execution continues, and the **Global Exception Handler** doesn't step in.