RPA Specialization

Course 4: Control Flow in RPA

Best Practices for Error Handling

Best Practices for Error Handling

Two types of exceptions may occur when running an automated process: predictable or unexpected. Based on this distinction, there are two ways of addressing exceptions, either by explicit actions executed automatically within the workflow or by escalating the issue to a higher level.

The best practices for handling errors are:

- Breaking the process into smaller workflows to better understand the code, independent testing, and reusability.
- Using Try/Catch blocks to predict and handle exceptions - Exception propagation can be controlled by placing susceptible code inside Try Catch blocks where situations can be appropriately handled. The main process diagram must define broad corrective measures at the highest level to address all generic exceptions and ensure system integrity. Enough details should be provided in the exception message for a human to understand it and take the necessary actions. The exception message and source are essential. The source property of an Exception object will indicate the name of the activity that failed (within an invoked workflow). Again, naming is vital - a poor naming practice will give no clear indication about the component that crashed or the source of the problem
- Using Global Exception Handler for situations that are global and/or less probable to happen.
- Using Rethrow - In some cases, it may be necessary to return the exception to the normal flow by using the Rethrow activity. This activity can only be used inside the Catches block of a Try Catch activity, and, as it does not receive any input, it uses the same exception that the Catches block caught. A common use for Rethrow is when catching an exception for a particular action (for example, logging) and rethrowing it for processing in upper levels