

## Control Flow in RPA

mercoledì 22 maggio 2024 12:23

## Sequences

A sequence is a container in which multiple activities are placed one after another and executed linearly.

- (1) It enables seamless movement from one activity to another
- (2) Activity is followed by the next activity in a linear fashion
- (3) Sequences can be reused
- (4) Sequence can contain any number of activities but no activity can be skipped during the execution and all activities execute in the order they are placed.

## Control Flow

It's the order in which activities or actions are executed or performed in a workflow.

It's used to define the decisions to be made during the execution of a workflow. These help you determine the path of a process execution based on a specific result.

The control flow activities can be found in the activities panel by navigating to the workflow category and then to the control category

There are two ways in which you can control the flow :

- (1) **Decision-based** : A decision is made based on a specific condition.

If the condition is met, the program executes one part of the workflow and if the condition is not met, then it executes the other part. (⇒ IF ; SWITCH)

- (2) **Iteration-based** : A particular part of a program is executed multiple times until a specific condition is met or holds true.

(⇒ While ; Do While ; For Each)

Other activities also help you control the flow of execution



## ① IF Activity

It contains a statement with a condition attached and two sets of instructions as outcomes. It's useful to make decisions

BASED ON THE VALUE OF VARIABLES

BASED ON THE VALUE OF EXPRESSION

## ② Switch Activity

It executes A SET OF STATEMENTS OUT OF MULTIPLE, BASED ON THE VALUE OF A SPECIFIC EXPRESSION.

It's USED IN PLACE OF AN IF ACTIVITY WHEN AT LEAST THREE POTENTIAL COURSES OF ACTIONS ARE NEEDED.

It CAN BE USEFUL TO CATEGORIZE DATA ACCORDING TO A CUSTOM NUMBER OF CASES.

	If Activity	Switch Activity
01	Suitable for taking a decision	Suitable for testing a given variable's value against a list of case values
02	Check for equality as well as logical expression	Check only for equality
03	Expression decides whether the statements inside 'Then' or 'Else' section is to be executed	Expression decides the case to be executed

## ③ Do While Activity

It executes the CONTAINED ACTIVITY FIRST AND THEN CHECKS IF THE CONDITION IS TRUE. THIS ALSO MEANS THAT IT WILL BE EXECUTED AT LEAST ONCE, EVEN IF THE CONDITION IS NOT TRUE.

## ④ While Activity

It executes the CONTAINED ACTIVITY WHILE THE PROVIDED CONDITION IS TRUE.

It CHECKS THE CONDITION AT THE BEGINNING BEFORE GETTING INTO THE LOOP.

If the CONDITION IS NOT TRUE, THE LOOP DOESN'T START.

## ⑤ For Each Activity

It ITERATES THROUGH THE ELEMENTS OF LISTS; ARRAYS; DATASETS, ... AND PERFORMS AN ACTION ON EACH ELEMENT INDIVIDUALLY

## Other Control Flow Activities

### (1) Delay Activity

It ENABLES YOU TO PAUSE THE AUTOMATION FOR A CUSTOM PERIOD IN HH:MM:SS FORMAT

### (2) Break Activity

It ENABLES YOU TO STOP THE LOOP AT A CHOSEN POINT AND THEN CONTINUE

WITH THE NEXT ACTIVITY.

THIS ACTIVITY CAN ONLY BE USED WITHIN "FOR EACH", "WHILE", "DO WHILE"

### (3) Assign Activity

It ASSIGNS A VALUE TO A VARIABLE OR AN OBJECT.

- ↳ (1) INCREMENT THE VALUE OF A VARIABLE IN A LOOP
- (2) SUM UP THE VALUE OF TWO OR MORE VARIABLES
- (3) ASSIGN THE RESULT TO A DIFFERENT VARIABLE
- (4) ASSIGN VALUES TO AN ARRAY

### (4) Continue Activity

It enables you to skip the REMAINING STEPS IN THE CURRENT ITERATION INSIDE A LOOP ACTIVITY (FOR EACH, WHILE, DO WHILE) AND CONTINUE THE EXECUTION WITH THE NEXT ITERATION.

### (5) Parallel Activity

It executes child activities in parallel. This means that by using

Parallel, multiple activities can be executed side by side.

It is helpful when there is the need to run several processes at the same time.

A Parallel Activity lets you schedule two or more child activity BRANCHES FOR PROCESSING SIMULTANEOUSLY.

However, it cannot run multiple processes in the foreground at the same time.

To run more than one process at the same time you must choose a single activity BRANCH to run in the foreground, AND the REMAINING ACTIVITY BRANCHES MUST RUN in the BACKGROUND.

↳ you can use "Send Message" or "Signal Type" PROPERTY of the activities for this purpose  
If they are not used, the Parallel Activity BEGINS PROCESSING with the execution of the activity BRANCH at a time.

It completes one activity BRANCH and then RANDOMLY PICKS another activity BRANCH.

It gives the result RANDOMLY when you finish this process in the PARENT ACTIVITY

## Flow CHARTS

A Flowchart is an activity that can act as a container for other activities

AND THESE ACTIVITIES CAN BE CONNECTED IN MULTIPLE WAYS.

It is a diagrammatical approach that represents various steps involved in completing tasks, activities and processes.

- (•) IT HELPS THE USER TO VIEW AND FOLLOW THE PROCESS QUICKLY
- (•) IT CAN BE EITHER USED AS STAND-ALONE AUTOMATION PROJECTS OR INCLUDED IN LARGER PROJECTS.

THE IMPORTANT ASPECT OF FLOWCHARTS IS THAT UNLIKE SEQUENCES, THEY PRESENT MULTIPLE BRANCHING LOGICAL OPERATORS THAT HELP CREATE COMPLEX BUSINESS PROCESSES AND CONNECT ACTIVITIES IN MULTIPLE WAYS.

### Decision - Making Activities

THERE ARE TWO ACTIVITIES USED TO DECIDE THE FLOW OF A PROGRAM INSIDE A FLOWCHART. THESE ARE:

#### (1) Flow Decision Activity

It's an activity that executes one of the two branches depending on whether a specified condition is met.

THIS ACTIVITY CAN ONLY BE USED IN A FLOWCHART AND IS EQUIVALENT TO THE IF ACTIVITY.

#### (2) Flow Switch Activity

It's an activity that splits the control flow into 3 or more branches. Out of these branches, only one is executed based on the specified condition.

THIS ACTIVITY CAN ONLY BE USED IN A FLOWCHART AND IS EQUIVALENT TO THE SWITCH ACTIVITY

### Loops in Flowcharts

Loops are structures used to automate repetitive tasks.

A PARTICULAR SECTION IN THE FLOWCHART MAY BE REQUIRED TO PERFORM REPETITIVE TASKS.

In flowcharts, the simplest types of loops can be created by connecting one activity to another. Generally, conditions are used in a flow decision activity within the flowchart to run a loop a specified number of times.

However you can create loops in flowchart without using any

## Nesting Flowcharts

Nesting is used while creating complex workflows as it allows logical division of the program and promotes reusability.

It is technically possible to nest flowcharts and sequences in every way. However, the only sustainable combination is to use a flowchart to represent the overall logic of the program and to use sequences for different parts and side

Therefore, it is recommended that you use sequences within the flowcharts and not the other way around.

## Errors AND Exceptions

An error is any type of event that prevents the normal execution of a program

Some of the errors:

- (1) Stop the execution of a process
- (2) Delay the execution of a process
- (3) Interfere with the execution without having a clear impact

1) **Syntax Error** = The compiler/interpreter cannot parse the written code into meaningful computer instructions.

2) **User Error** = where the software determines that the user's input is not acceptable for some reason

3) **Programming Error** = where the program contains no syntax errors but it does not produce the expected results (= bugs)

**Exceptions** = They are a subset of errors that are serious enough to produce a notable effect and for which there can be a mechanism to identify and address them.

**Exception Handling Mechanism** refers to:

- 1) How to prevent and/or respond to exceptions
- 2) Sometimes the handling mechanism can be simply stopping the execution.

**Types of Exceptions:**

(1) **Application or System Exception**

It describes an error rooted in a technical issue, such as an application that is not responding.

This type of error has a chance of being solved simply by retrying the transaction, as the application can unpickle.

It can be managed by following good naming conventions for activities and workflows. This helps in tracking the activity that caused the exception.

## G) Business Exception

It describes an error rooted in the fact that certain data which the automation project depends on is incomplete, missing, outside of set boundaries, or does not pass other data validation criteria.

It is an exception from the usual process flow, and the validation is made explicitly by the developer inside the workflow.

The text in the exception should contain enough information for a human user to understand what happened and what action needs to be taken.

## Error Handling Approach

It is common for automation projects to encounter events that interrupt or interfere with the projected execution.

The different approaches are:

- (1) Stopping the execution
- (2) Executing specified actions within the workflow
- (3) Escalating the issue to human operation

Studio offers 3 activities to handle errors:

- (1) Try Catch
- (2) Retry Scope
- (3) Global Exception Handler

## Try Catch

It's used when specific parts of the project may trigger errors.

The activity catches a specified exception type in a sequence and either displays an error notification or dismisses it and continues the execution.

The activity has 3 main sections:

(1) **Try** : THIS HOLDS THE ACTIVITY THAT COULD THROW AN EXCEPTION

(2) **Catches** : IT SPECIFIES AN ACTIVITY OR A SET OF ACTIVITIES TO BE PERFORMED WHEN AN ERROR OCCURS.

IT HAS AN "EXCEPTION" SECTION WHERE THE EXCEPTION TYPES ARE DEFINED, HERE MULTIPLE EXCEPTIONS CAN BE ADDED.

(3) **Finally** : IT HOLDS AN ACTIVITY THAT WILL BE EXECUTED ONLY IF NO ERROR OCCURS OR IF THE ERROR IS ALREADY CAUGHT

## Retry Scope

IT RETRIES THE CONTAINED ACTIVITY AS LONG AS THE CONDITION IS NOT MET OR AN ERROR IS THROWN

(1) IT IS USED TO RETRY THE EXECUTION IN SITUATIONS WHERE AN ERROR IS EXPECTED. THE EXECUTOR IS RETRIED UNTIL A CERTAIN EVENT HAPPENS A NUMBER OF TIMES OR WITHOUT ANY CONDITION WHERE IT IS RETRIED OR NO EXCEPTION IS THROWN.

(2) IT IS USED FOR CATCHING AND HANDLING AN ERROR, WHICH IS WHY IT'S SIMILAR TO "TRY CATCH"

THE 2 MAIN PROPERTIES ARE:

- 1) **Number Of Retries** : NUMBER OF TIMES THE SEQUENCE IS TO BE RETRIED
- 2) **Retry Interval** : SPECIFIES THE AMOUNT OF TIME TO WAIT BETWEEN EACH RETRY

## Continue on Error Property

IT'S A PROPERTY FOUND IN THE ACTIVITIES IN STUDIO. IT'S USEFUL FOR ACTIVITIES THAT WORK WITH UI INTERACTIONS.

IT'S THE EASIEST WAY TO INSTRUCT THE WORKFLOW TO CONTINUE, EVEN IF THE ACTIVITY THROWS AN ERROR

THIS PROPERTY ONLY SUPPORTS BOOLEAN VALUES. IF "True", IT CONTINUES THE EXECUTION EVEN IF THERE ARE SOME ERRORS.

(1) IT'S USED WHEN USING DATA SCRAPING, SO THAT THE ACTIVITY DOESN'T THROW AN ERROR ON THE LAST PAGE.

(2) IT'S USED WHEN THE USER IS NOT INTERESTED IN CAPTURING THE ERROR, BUT SIMPLY IN THE EXECUTION OF THE ACTIVITY.

