

# UiPath Studio Activities Used with Queues

## Queue

A Queue is a container that enables you to hold an unlimited number of items. Queue items can store multiple types of data. When you create a queue in Orchestrator, it is empty. To populate the queue with items, change their status and process them, you have to use activities from Studio.

## Package:

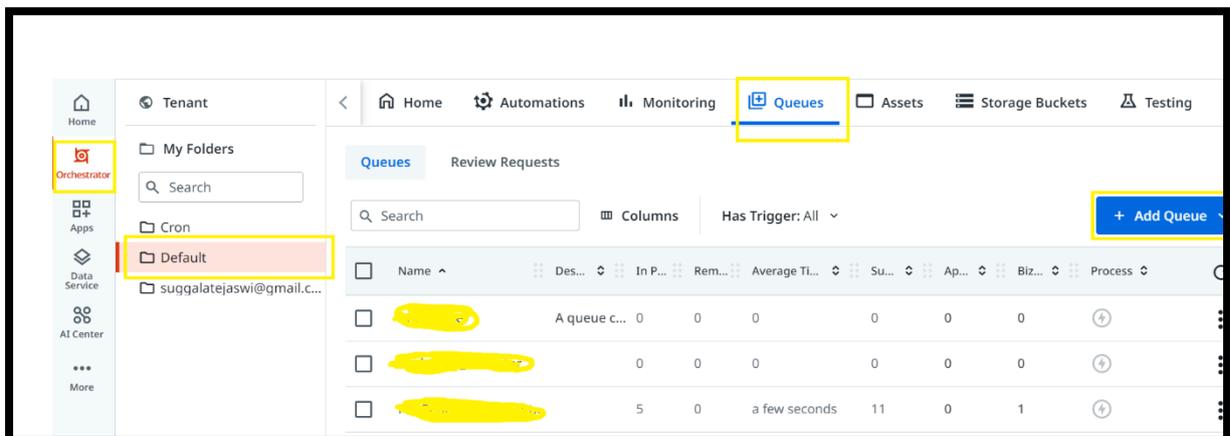
Queue activities available in under **UiPath.System.Activities** package.

## Pre-Requisite:

To work with these activities the Orchestrator should be connected with the Robot.

## How to create queue in orchestrator:

- 1.Login into [cloud.uipath.com](https://cloud.uipath.com)
- 2.Click on orchestrator and select folder, go to queue section and click on Add Queue select Create a New Queue.



- 3.Give the queue name and description.We can choose unique reference,Auto retry,Max of retries then click on add.

**Name:** Mandatory.The name of the queue. It's better to give process name as queue name.

**Description:** Optional.The description of your queue.

**Unique Reference:** Optional. Enables to enforce unique references in all the transactions that are to be added to the queue, by selecting the Yes button. If the No button is selected, then the queue can contain transactions with duplicate references.

This option cannot be edited, and displays if the current queue enforces unique references or not.

**Auto Retry:** Mandatory. Indicates if Orchestrator should retry to process items in the queue when an Application Exception is encountered. Yes or No are the available options. If you select No, the Max # of retries field is no longer displayed.

**Max # of Retries:** Mandatory. The number of times Orchestrator should retry to process an item when an Application Exception is encountered. By default, the value is 1. The maximum number of retries is 50.

**Specific Data JSON Schema:** Optional. File size limited to 256kb by default. Enables you to upload a JSON schema for the specific data of all transactions in the queue.

All transactions added to the queue must adhere to the schema or an error will be triggered and the request to add the transaction will fail.

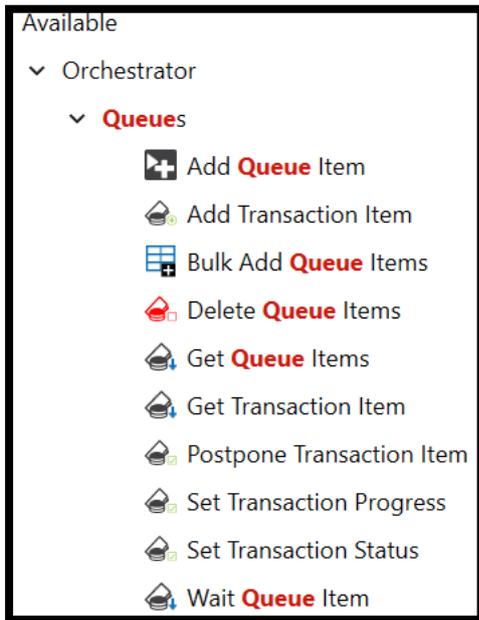
Note: The uploaded schema is not enforced retroactively, existing transaction items are not affected.

**Output Data JSON Schema:** Optional. File size limited to 256kb by default. Enables you to upload a JSON schema for the output data of all transactions in the queue. If a transaction status is set to Successful but its output data violates the schema, then the transaction will fail with a Business Exception.

**Analytics Data JSON Schema:** Optional. File size limited to 256kb by default. Enables you to upload a JSON schema for the analytics data generated when a transaction status is set to Successful. If a transaction status is set to Successful but its analytics data violates the schema, then the transaction will fail with a Business Exception.

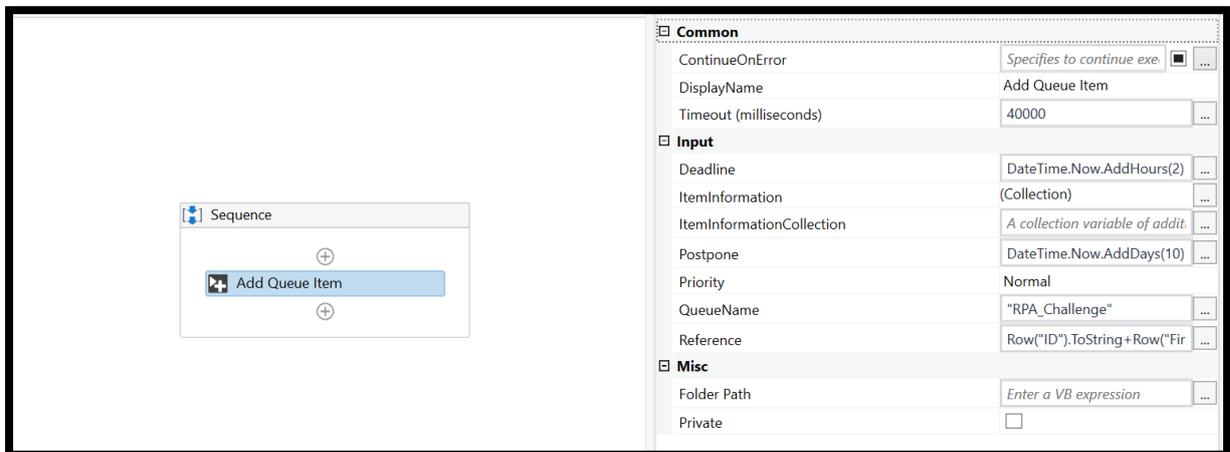
Name *	Specific Data JSON Schema
<input type="text" value="HR-GenerateOfferLetter"/>	<input type="text"/> <a href="#">Browse</a>
Description	Output Data JSON Schema
<input type="text" value="Generating offer letters for hired candidates"/>	<input type="text"/> <a href="#">Browse</a>
Unique Reference	Analytics Data JSON Schema
<input type="radio"/> Yes <input checked="" type="radio"/> No	<input type="text"/> <a href="#">Browse</a>
Auto Retry	
<input checked="" type="radio"/> Yes <input type="radio"/> No	
Max # of retries	
<input type="text" value="1"/>	
<input type="button" value="Cancel"/> <input type="button" value="Add"/>	

## UiPath Studio Queue Activities:



### Add Queue Item:

populates the queue with items, defines a time frame when these items must be processed, and adds a custom reference to each item. By default, when an item is added to the queue its status is **New**.



## Properties:

Properties

UiPath.Core.Activities.AddQueueItem

**Common**

ContinueOnError *Specifies to continue exe*  ...

DisplayName Add Queue Item

Timeout (milliseconds) 40000 ...

**Input**

Deadline DateTime.Now.AddHours(2) ...

ItemInformation (Collection) ...

ItemInformationCollection *A collection variable of addit* ...

Postpone DateTime.Now.AddDays(10) ...

Priority Normal

QueueName "RPA\_Challenge" ...

Reference Row("ID").ToString+Row("Fir" ...

**Misc**

Folder Path *Enter a VB expression* ...

Private

**QueueName:** The queue where the QueueItem object is to be added. The name is case insensitive.

**ItemInformation:** A collection of additional information about the specific QueueItem that is to be added. The information is stored in the item and is used during a transaction processing.

Name	Direction	Type	Value
FirstName	In	String	Row("FirstName").ToString
LastName	In	String	Row("LastName").ToString
Role	In	String	Row("Role").ToString
Company	In	String	Row("Company").ToString

OK Cancel

**Priority:** The priority level of the queue item to be added. This property is a criterion for the prioritization of queue items, alongside Deadline and Postpone.

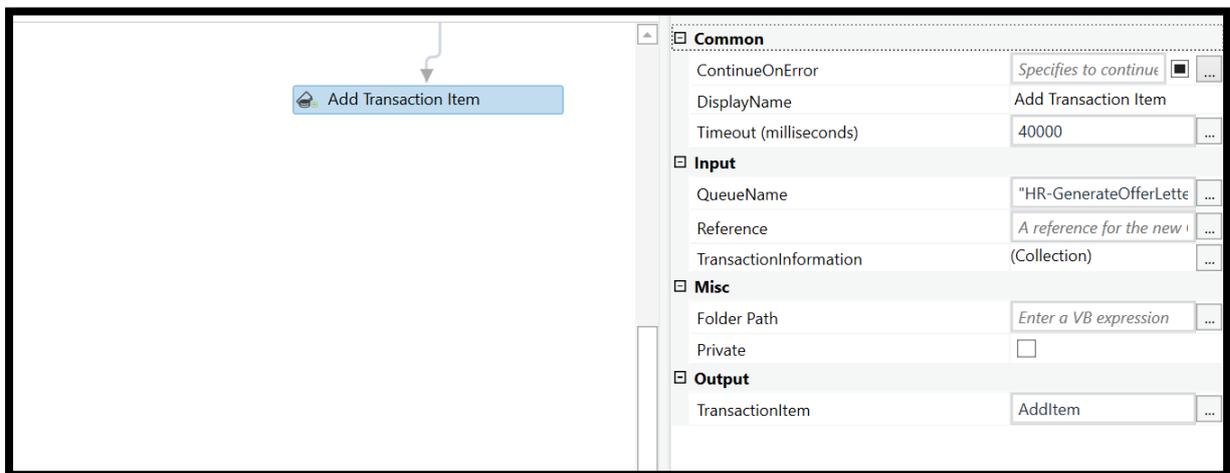
**Postpone:** The date after which the queue item may be processed. This property is a criterion for the prioritization of queue items, alongside Priority and Deadline.

**Deadline:** The date before which the queue item should be processed. This property is a criterion for the prioritization of queue items, alongside Priority and Postpone.

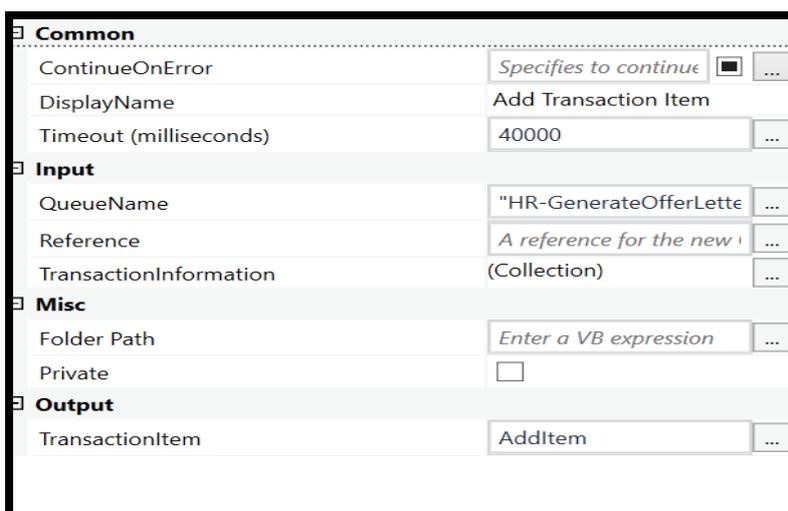
**Reference:** The reference of the added QueueItem. The reference can be used to link your transactions to other applications used within an automation project. Additionally, this feature enables you to search for certain transactions, in Orchestrator, according to the provided string.

## Add Transaction Item:

Adds an item in the queue, starts the transaction and sets the status to **In Progress**. It can also be configured to add a custom reference to each transaction. Returns the item as a QueueItem variable.



## Properties:



### Properties: Input

**TransactionInformation:** A collection of additional information about the specific TransactionItem that is to be added. The information is stored in the item and is used during a transaction processing.

Name	Direction	Type	Value
First Name	In	String	"[REDACTED]"
Last Name	In	String	"[REDACTED]"
Id	In	String	"[REDACTED]"

Create Argument

OK Cancel

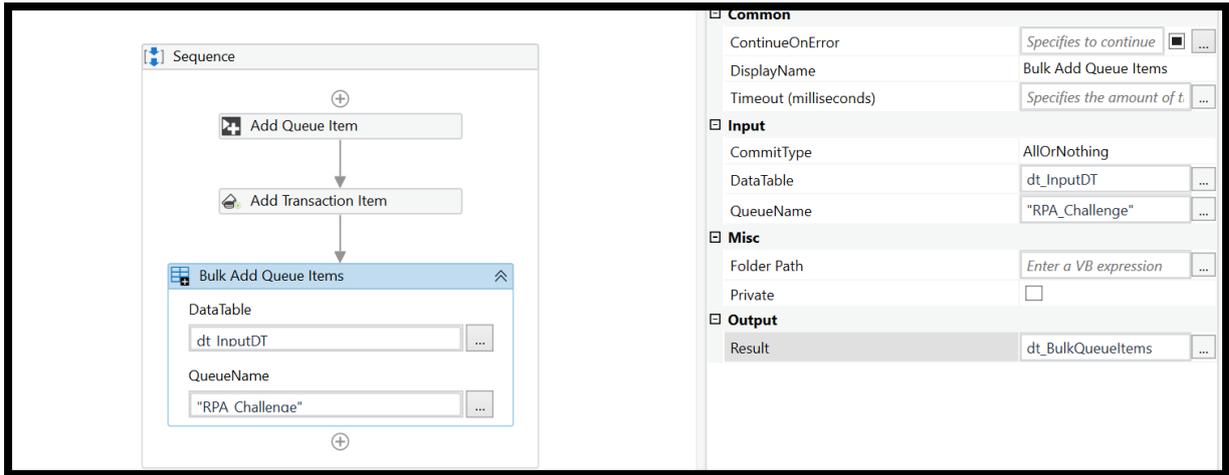
**QueueName:** The queue where the QueueItem object is to be added. The name is case insensitive.

### Properties: Output

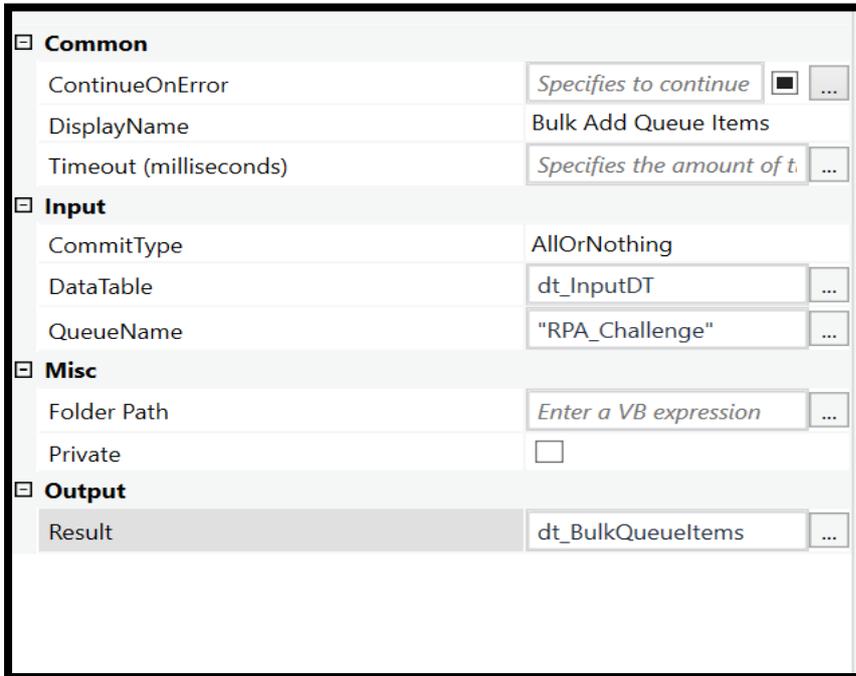
**TransactionItem:** The added TransactionItem.

### Bulk Add Queue Items:

Adds a collection of items from a specified DataTable to a specified queue and sets the items' statuses to New.



## Properties:



## Properties: Input

**CommitType:** A drop-down which specifies the way the items are added to the queue, as well as the manner in which the errors are treated. Available options are:

- **AllOrNothing:** The activity adds all of the specified queue items if no error occurs. In the case of an error, no items are added and the row where the error occurs is returned.
- **ProcessAllIndependently:** Adds each queue item individually and returns a list of all queue items that could not be added, if any.

**DataTable:** The DataTable variable that contains the values you want to add to the specified queue. This field supports only DataTable variables.

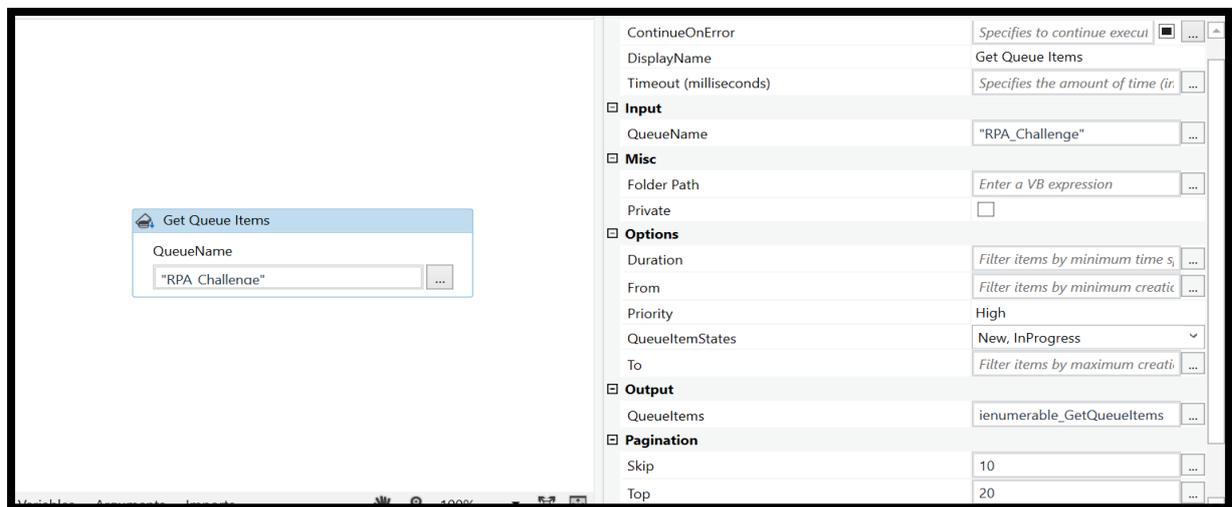
**QueueName:** The name of the queue where you want to add the new queue items. This field supports only strings and String variables.

### Properties: Output

**Result:** Depending on the selection made in the CommitType property, returns all errors that occurred during the process, storing them in a DataTable variable. This field supports only DataTable variables.

### Get Queue Items:

Retrieves a list of up to 100 transactions from an indicated queue, according to multiple filters, such as creation date, priority, state, and reference.



## Properties:

The screenshot shows the 'UiPath.Core.Activities.GetQueueItems' properties window. It is organized into several expandable sections:

- ContinueOnError:** A checkbox with the label 'Specifies to continue exe' and a '...' button.
- DisplayName:** A text field containing 'Get Queue Items'.
- Timeout (milliseconds):** A text field with the label 'Specifies the amount of time' and a '...' button.
- Input:** A section containing 'QueueName' with a text field containing '"RPA\_Challenge"' and a '...' button.
- Misc:** A section containing 'Folder Path' with a text field containing 'Enter a VB expression' and a '...' button, and 'Private' with an unchecked checkbox.
- Options:** A section containing 'Duration' with a text field containing 'Filter items by minimum tim' and a '...' button, 'From' with a text field containing 'Filter items by minimum crec' and a '...' button, 'Priority' with a dropdown menu set to 'High', 'QueueItemStates' with a dropdown menu set to 'New, InProgress', and 'To' with a text field containing 'Filter items by maximum cre' and a '...' button.
- Output:** A section containing 'QueueItems' with a text field containing 'IEnumerable\_GetQueueItem' and a '...' button.
- Pagination:** A section containing 'Skip' with a text field containing '10' and a '...' button, and 'Top' with a text field containing '20' and a '...' button.

### Properties:Input

**QueueName** - The name of the queue we want to extract data from. The name has to be the same as the one in Orchestrator.

### Properties: Options

**Duration:** Filters transactions by the minimum time they spent in the previous attempt.

**From:** Filters transactions by the minimum creation date. It accepts DateTime values.

**Priority:** Filters transactions according to their priority. This field supports only integers and integer variables. The priorities are:

- 0 - High;
- 1 - Normal;
- 2 - Low.

**QueueItemStates:** Filters The queue item states. The available options are:

All, New, InProgress, Failed, Successful, Abandoned, Retried, Deleted

**To:** Filters transactions according to the maximum creation date. It accepts DateTime values.

### Properties:Output:

**QueueItems:** The list of items that you want to mark as retrieved from Orchestrator. This property supports only IEnumerable variables.

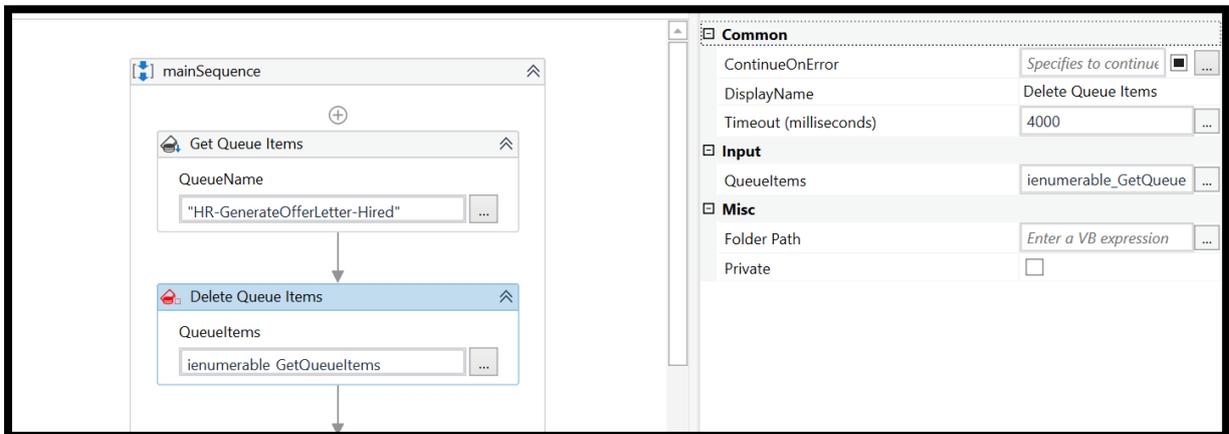
## Pagination:

**Skip:** Helps you skip the first "n" items. This property supports only integers and integer variables.

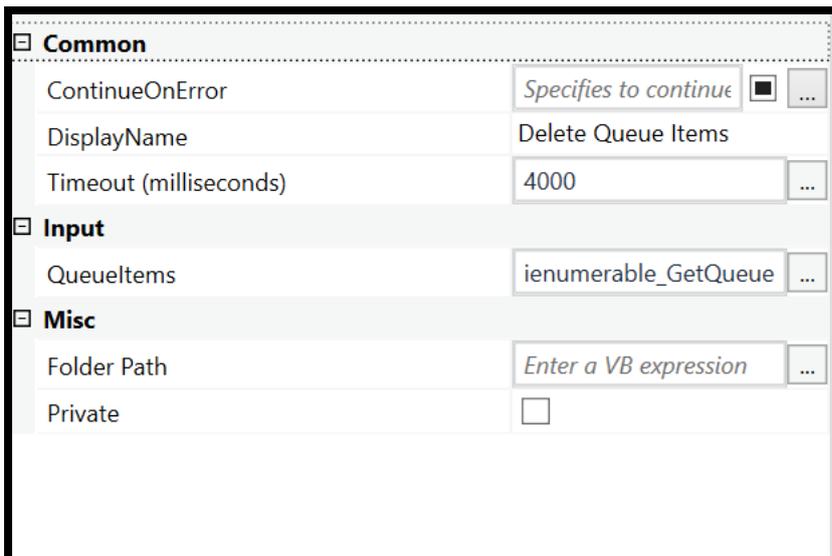
**Top:** Returns only a specified number of items out of the total. The maximum value is 100. This property supports only integers and integer variables.

## Delete Queue Items:

Deletes items with the **New** status from a specified queue. In order to execute this activity in Studio, the Robot has to be connected to Orchestrator and the Robot role needs Edit, Create, and Delete permissions on queues and Delete permissions on transactions.



## Properties:



## Properties:Input

**QueueItems:** The list of items that you want to mark as deleted in Orchestrator. They must be in the 'New' state and should be retrieved via a **Get Queue Items** activity first. This property supports only IEnumerable variables.

## Get Transaction Item:

Gets an item from the Orchestrator queue so that you can process it (start the transaction) and sets its status to In Progress. Starting its processing makes it a transaction. To execute this activity, the Robot role needs to have Edit permissions on queues. The activity runs under the Robot which executes it.

## Properties: Input

**QueueName:** The queue where the QueueItem object is to be added. The name is case insensitive.

The screenshot displays the Microsoft Orchestrator interface. On the left, a workflow diagram is visible within a window titled 'Queues'. The workflow starts with a 'Get Transaction Item' activity, followed by two 'Assign' activities. The first 'Assign' activity sets 'str\_FirstName' to 'qitem\_GetTransacti'. The second 'Assign' activity sets 'str\_LastName' to 'qitem\_GetTransacti'. The workflow then proceeds to a 'Write Line' activity with the text 'str\_FirstName+str\_LastName'. On the right, the properties pane for the 'Get Transaction Item' activity is shown. It includes sections for 'Common', 'Input', 'Misc', 'Output', and 'Reference'. The 'Input' section shows 'QueueName' set to 'HR-GenerateOfferL'. The 'Output' section shows 'TransactionItem' set to 'qitem\_GetTransactio'. The 'Reference' section shows 'FilterStrategy' set to 'StartsWith' and 'Reference' set to 'Filter items by referei'.

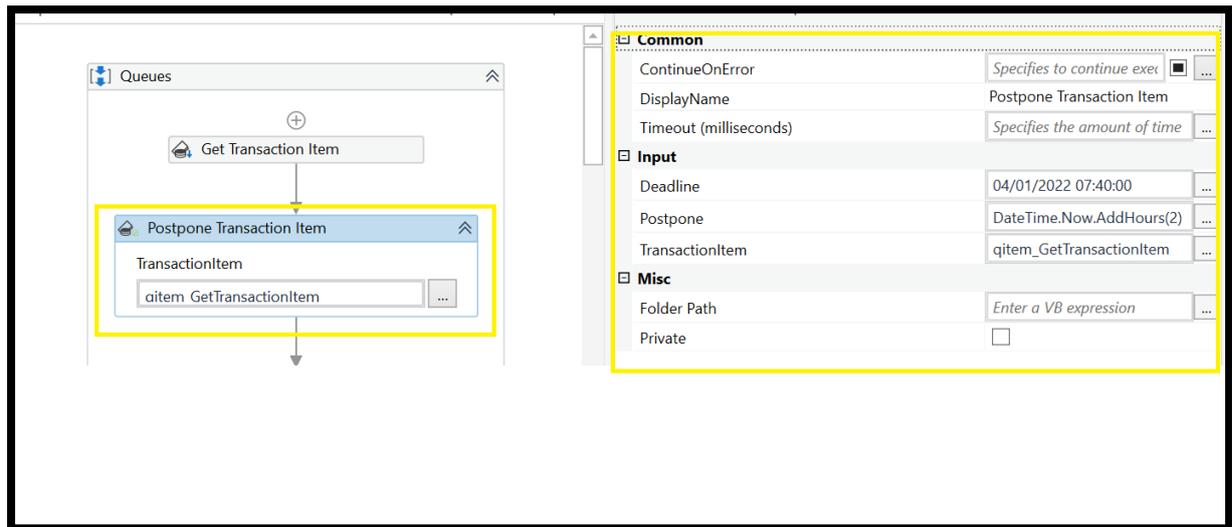
## Properties: Output

**TransactionItem:** The retrieved Transaction Item, stored in a QueueItem variable.

Retrieving for  
FirstName=>str\_FirstName=qitem\_GetTransactionItem.SpecificContent("FirstName").  
ToString

## Postpone Transaction Item:

Adds time parameters between which a transaction must be processed (not before **Postpone** and no later than **Deadline**).



## Properties:Input

**Postpone:** The date after which the queue item may be processed. This property can be filled in with relative dates such as `DateTime.Now.AddHours(2)`, `DateTime.Now.AddDays(10)` and `DateTime.Now.Add(New System.TimeSpan(5, 0, 0, 0))`.

**Deadline:** The date before which the queue item should be processed. This property can be filled in with relative dates such as `DateTime.Now.AddHours(2)`, `DateTime.Now.AddDays(10)` and `DateTime.Now.Add(New System.TimeSpan(5, 0, 0, 0))`.

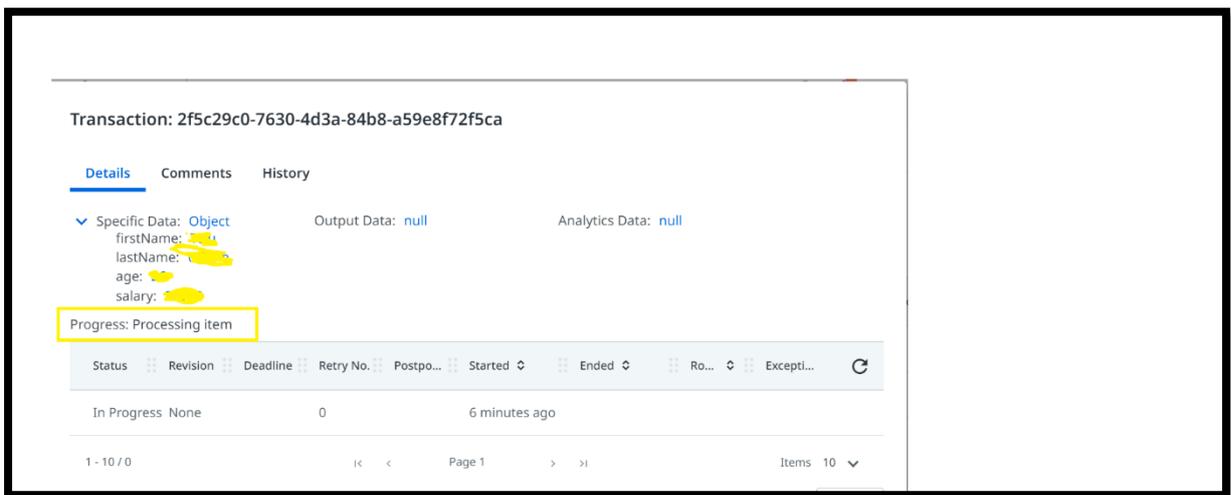
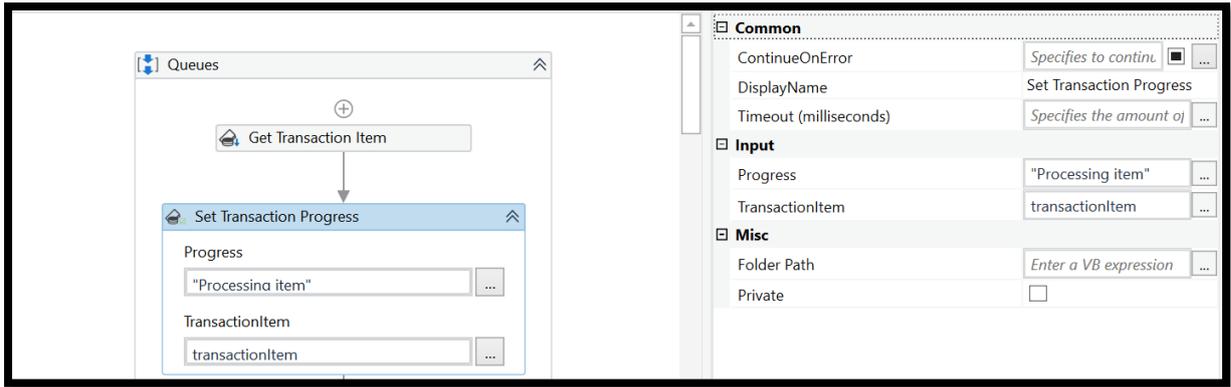
**TransactionItem:** The TransactionItem to be postponed.

## Set Transaction Progress:

Helps to create custom progress statuses for **In Progress** transactions. This can be placed in key places in automation project, to inform its progress if, for example, it crashes. This option can be useful in the troubleshooting process.

**SetTransactionProgress** Activity is used to set the custom progress for **"InProgress"** status in Queue.

SetTransactionProgress activity updates the progress of your bot flow.



## Properties:Input

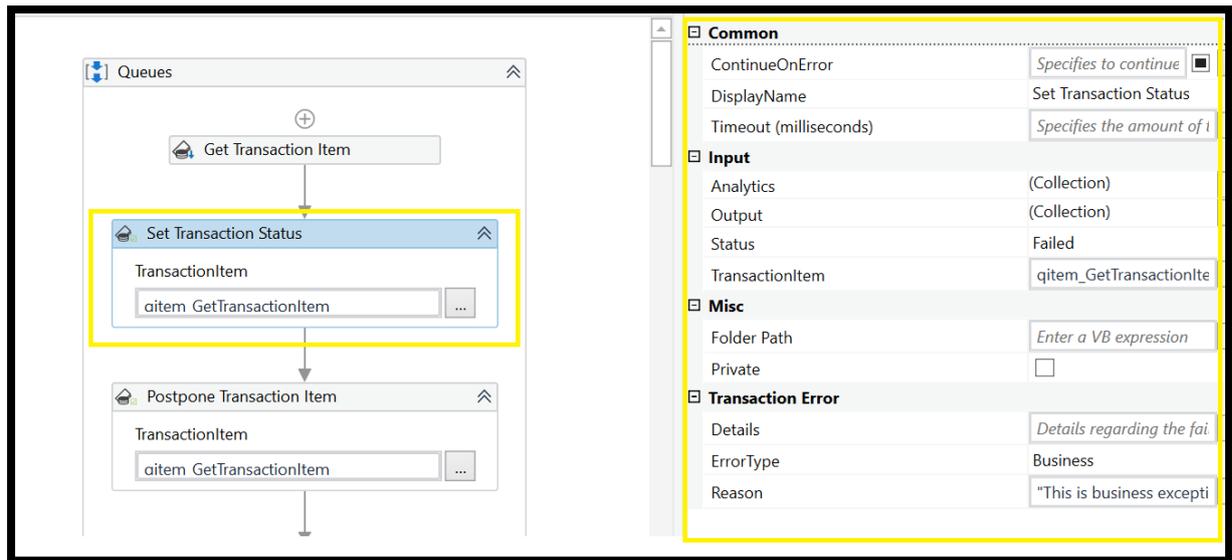
**Progress:** The custom progress status that is to be set to the TransactionItem. This field supports only strings and string variables.

**TransactionItem:** The TransactionItem whose progress is to be updated.

## Set Transaction Status:

Changes the status of the transaction item to Failed (with an application or Business Exception) or Successful.

Application Exceptions should be used when an app you are automating fails or is not responding, while Business Exceptions should be used when a certain transaction value does not meet a business requirement.



### Properties: Input

**Analytics:** A collection of analytics information about the specific TransactionItem whose status is to be updated.

**Output:** A collection of additional information about the specific TransactionItem whose status is to be updated.

**Status:** The status that is to be set to the TransactionItem.

**TransactionItem:** The TransactionItem whose status is to be updated.

### Properties: Transaction Error

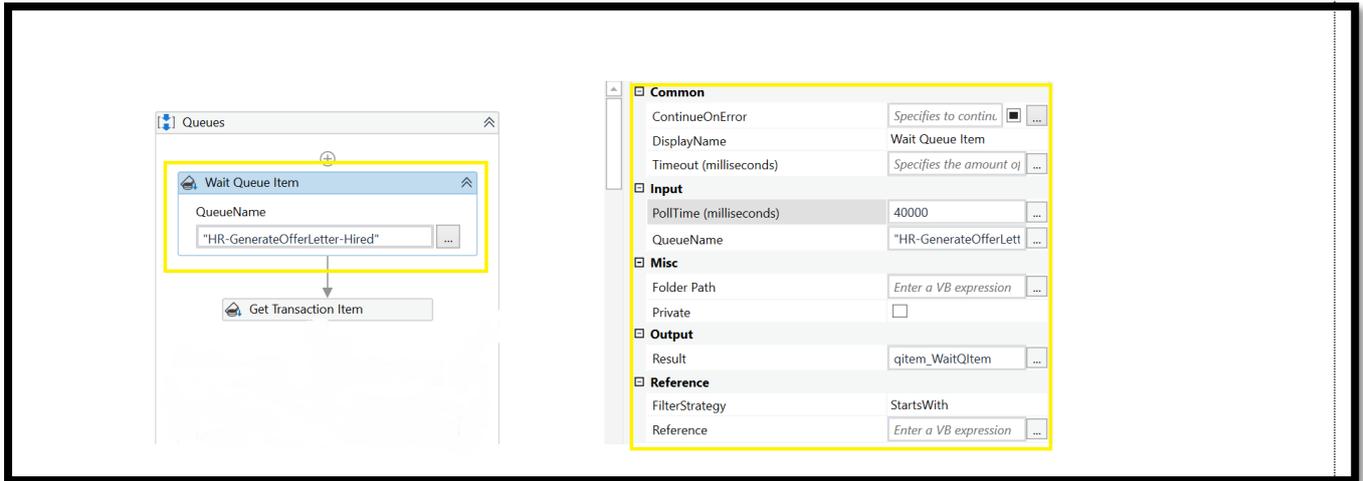
**Details:** Details regarding the failed Transaction. This field supports only strings and String variables.

**ErrorType:** The error type that the failed Transaction has thrown.

**Reason:** The reason for which the Transaction failed. This field supports only strings and String variables.

### Wait Queue Item:

Wait for an item in the queue to be available so that we can process it and sets its status to **In Progress**.



### Properties: Input

**PollTime (milliseconds):** Specifies the amount of time (in milliseconds) to wait before looking again for the queue item. The default value is 30000 milliseconds (30 seconds).

**QueueName:** The name of the queue where you want to search for the queue item. This field supports only strings and String variables.

### Properties: Output

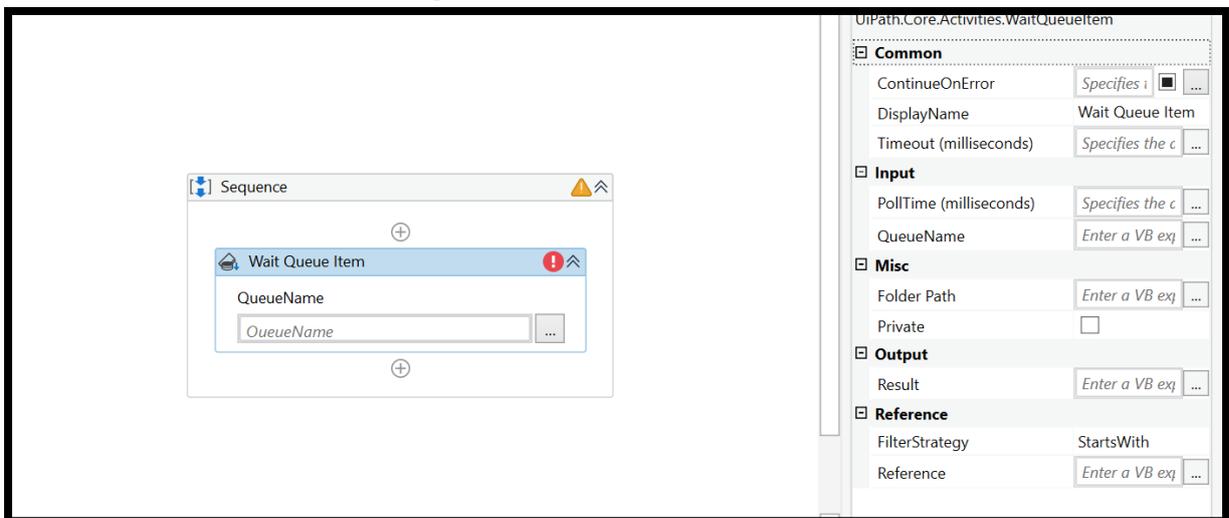
Result - The retrieved queue item, stored in a QueueItem variable.

### Properties: Reference

**FilterStrategy:** The strategy to use when filtering according to reference. Available options are:

- **StartWith:** Looks for items whose references start with the value indicated in the Reference field.
- **Equals:** Looks for items whose references are the same as the value indicated in the Reference field.

**Reference:** Filters items according to their reference.



## Queue Item Statuses

These statuses let you know if an item has been processed or not, and the stage of the process at a particular time. Item statuses are displayed in the Status column, in the Transactions page. Queue Item statuses are:

- **New:** The item has just been added to the queue with the Add Queue Item activity, or the item was postponed, or a deadline was added to it, or the item was added after an attempt and failure of a previous queue item with auto-retry enabled.
- **In Progress:** The item was processed with the Get Transaction Item or the Add Transaction Item activity; when an item has this status, custom progress status is also displayed, in the Progress column.
- **Failed:** The item did not meet a business or application requirement within the project and sent to a Set Transaction Status activity, which changed its status to Failed.
- **Successful:** The item was processed and sent to a Set Transaction Status activity, which changed its status to Successful.
- **Abandoned:** The item remained in the In Progress status for a long period of time (approx. 24 hours) without being processed;
- **Retried:** The item failed with an application exception and was retried. After the Robot finishes retrying the item, the status changes to Failed or Successful, according to workflow.
- **Deleted:** The item has been manually selected from the Transactions page and marked as deleted. An item with this status can no longer be processed.