What is Orchestrator?

what is user

what is role

what is Assest?

waht s package

what is Process

what is backoffice and front office robot

What is Queues?

What is High density robots?

Can i map 1 robots in two different envirionment

how to update packages?

How to schedule the bots.

If 1 process is map with let say 'A' envirionment can we map same process to 'B' Envirionment?

Orchestrator

it is used deploy process and add robots and also manage them well. It provide role, user process , package , envoirment , job , schedules options

What is tenent?

The Default tenant is also automatically created when installing Orchestrator,

as your first tenant. Each tenant has a tenant admin, created as a local user, and has the Administrator role.

The tenant admin cannot be deleted or disabled, but you can edit its information in the Users page.

Users

A user is an entity that stores the assigned role(s), email settings and enables you to login to Orchestrator. A

user’s view of Orchestrator is dependant on the assigned roles.

The **Users** page displays all available users, enables you to edit their details or remove them, and import users

from Active Directory groups.

Robot users can also be displayed in this page, yet you can only edit their roles. (A Robot user is automatically

created when you register a Robot to Orchestrator.)

Roles

The **Roles** page enables you to manage user permissions in Orchestrator. A user’s view of Orchestrator is

dependant on the role(s) assigned to it.

By default, the **Administrator** and **Robot** roles exist. They cannot be changed or removed.

A role enables you to administer view, edit, create and delete permissions to all Orchestrator pages and

components.

Robots

The **Robots** page enables you to provision robots and edit them, as well as view their status and the

environment(s) they are assigned to.

A Robot (Back or Front Office) is an execution host that runs processes built in UiPath Studio. From

Orchestrator, you can only run processes on Back Office Robots.

Any given Robot can have one of the following status:

● **Available** - the Robot is not running a process and is free to be used;

○ if this icon is displayed next to this status, it means that the Robot is **Unresponsive** , or that

the Robot and Orchestrator have not communicated in the last two minutes;

● **Busy** - the Robot is running a process;

○ if this icon is displayed next to this status, it means that the Robot is **Unresponsive** , or that

the Robot and Orchestrator have not communicated in the last two minutes;

**● Disconnected** - the UiPath Robot service is not running.

By default, only the **Administrator** has the right to register new Robots to Orchestrator.

Any given Robot can have one of the following status:

● **Available** - the Robot is not running a process and is free to be used;

○ if this icon is displayed next to this status, it means that the Robot is **Unresponsive** , or that

the Robot and Orchestrator have not communicated in the last two minutes;

● **Busy** - the Robot is running a process;

○ if this icon is displayed next to this status, it means that the Robot is **Unresponsive** , or that

the Robot and Orchestrator have not communicated in the last two minutes;

**● Disconnected** - the UiPath Robot service is not running.

Choose the type of Robot to use. The available options are:

● DevTest - this is the default option.

● Back Office

● FrontOffice

Environments

An environment is a grouping of Robots, that is used to deploy processes.

The **Environments** page displays all previously created environments and enables you to manage Robots

within them.

Select **Dev** , **Test** or **Prod** from the **Type** drop-down list, to indicate how the environment should be used.

Processes

The **Processes** page enables you to deploy an uploaded package to Robot environments, manage previously

created associations and keep all your processes up to date. This helps you distribute packaged on the Robot

machines and execute processes faster from the **Jobs** page .

Each time a package is linked to an environment, it is automatically distributed to all the Robot machines that

belong to that environment, in the %ALLUSERSPROFILE%\UiPath\Projects folder. When you update or

rollback a process , it is also automatically updated in the aforementioned directory.

When a new version of a package is available in Orchestrator (you published a new version of it from Studio), it

is indicated with an exclamation mark icon next to the process it is part of. For more information, see

Managing Process Versions .

If a package version associated with an environment is no longer available in the configured NuGet repository,

it is indicated with the following icon .

Assets

Assets enable you to store specific information so that the robots can easily have access to it. Additionally, all

credentials stored here are encrypted with the AES 256 algorithm.

The **Assets** page contains shared variables or configurations used by Robots in processes.

Assets can be requested by a workflow in Studio, through two activities: **Get Credential** and **Get Asset** .

The **Get Asset** and **Get Credential** activities from Studio request information from Orchestrator about a

specific asset, according to a provided **AssetName** .

There are four types of assets:

● **Text** - stores only strings (it is not required to add quotation marks)

● **Bool** - supports true or false values

**● Integer** - stores only whole numbers

● **Credential** - can contain usernames and passwords that the robot requires to execute particular

processes, such as login details for SAP or SalesForce.

Additionally, assets can be:

● **Global** - can be accessed and used by all Robots

● **Per Robot** - can be accessed only by a selected robot (created only by selecting the **Per Robot** check

box)

Queues

A queue is a list of items that you want to be processed by Robots. 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.

The **Queues** page enables you to create queues, view information about queue items (such as average

execution time and the total number of successful transactions), display charts with the transaction status

progress over time.

The following activities from Studio are used with queues:

● **Add Queue Item** - populates the queue with items and defines a time frame when these items must be

processed. By default, when an item is added to the queue its status is **New** .

● **Get Transaction Item** - gets an item from the queue so that you can process it (start the transaction)

and sets its status to **In Progress** .

● **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 the app you are

automating fails, while Busines

Transactions

The **Transactions** page displays all the transactions from the database, their statuses , the Robot that

processed them and the queue it belongs to. You can also manually mark them as verified or reviewed.

Queue items can have two types of statuses:

● **Item statuses** - let you know if the item has been processed or not, and the stage of the process at a

particular time. They are displayed in the **Status** column (in the items view of the **Queue** page). Queue

items can go through the following statuses:

○ **New** - the item has just been added to the queue with the **Add Queue Item** activity;

○ **In Progress** - the item was processed with the **Get Transaction Item** or **Add Transaction Item**

activities;

○ **Failed** - the item did not meet some business or application requirements within the project and

therefore, was sent to a **Set Transaction Status** activity that changed its status to Failed;

○ **Successful** - the item was processed and sent to a **Set Transaction Status** activity that 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 it was retried. After the Robot

finished retrying the item, the status changes to Failed or Successful, according to your

workflow.

● **Revision statuses** - let you perform version control, but **only of queue items that fail with an application**

**exception** . These statuses have to be manually set per item and have no implications in Orchestrator or

Studio, other than changing the value in the **Revision** column from the **Queues** page. The following

statuses are available:

○ **None** - this is the default status and it is set to all items, even if they failed or not

○ **In Review** - a user has marked an item that has failed with app exception as in the process of

being reviewed

○ **Verified** - a user has marked that the item has been verified (you cannot retry it after setting this

status)

○ **Retried** - the item has been marked manually for retry

Jobs

A job is a scheduled or manual execution of a process on a Robot.

The **Jobs** page displays all manually e

Schedules

The **Schedules** page displays all previously created schedules and enables you to schedule the execution of

processes at regular intervals, on selected Robots, all of them or a specified number of Robots. You can stop a

scheduled job after a custom amount of time, and manually enable or disable it.xecuted or scheduled jobs in reverse chronological order (from latest to

oldest). You can also to manually execute processes assigned to Robots, as well as cancel or terminate them.

**There is a need to understand the difference between Front Office and Back Office Robots. There is mainly two reasons why this is important and in this article I will explain why it is important.**

The two reasons why there is a need to understand the difference are the following are:

* They solve two different kinds of processes – so will you reduce cost or increase customer satisfaction?
* There can be an extra license impact involved using Front Office Robots

**What is the main difference between Front Office and Back Office Robot**

As they solve and have two different focus areas then it is good to have a basic overview of the main differences.

**Back Office Robots:**

* Doesn’t need human intervention
* Server side initiation of Robots
* Triggered by a scheduled or monitoring such as a file folder or inbox in web mail

**Front Office Robots:**

* Needs human intervention
* Standalone
* Triggered by a user and runs only under human supervision

-------------------------------------------------------------------------------------------------------------------------