# Orchestrator Manager

# Contents

# Overview

As RPA implementations scales, there are situations that require the manipulation of a large number of Orchestrator entities. Although Orchestrator's web interface provides a user-friendly way to see, create, edit and delete such entities, it can become a limiting factor if many operations need to be done in a short amount of time.

Orchestrator Manager is a tool that leverages Orchestrator's API to manipulate entities (Assets, Environments, Machines, Processes, Users, Robots, Organization Units, Folders, Queues and Packages) with data defined in Excel workbooks. It provides flexible and easy bulk operations on entities, enabling efficient solutions for situations that include:

- Registering a large quantity of users that obtain access to Orchestrator, including the specification of different roles and assignment to different Folders (Organization Units).
- Migration between tenants, including tenants hosted on UiPath Automation Cloud.
- Setting temporary passwords for a group of users according to security policies.
- Migration from Classic Folders to Modern Folders.

In addition, since Orchestrator Manager is offered as a UiPath Studio project, it can be customized and extended according to needs other than the ones above.

The target users of Orchestrator Manager are Orchestrator administrators with the required permissions and familiar with an Orchestrator instance's features, settings and data. By leveraging this tool, it is expected that Orchestrator administrators can decrease the time spent in regular data manipulation tasks and increase their overall efficiency.

The remaining of this guide is structured as follows: Section How to Use contains usage instructions, and the Architecture section details the main components of Orchestrator Manager. Limitations of the current version are listed in section Restrictions. Lastly, section Distribution and Support provides information about distribution, licensing and support.

# How to Use

The current version of Orchestrator Manager can be used with Orchestrator 2018.4, Orchestrator 2019.4, Orchestrator 2019.10, Orchestrator 2020.4 and Orchestrator instances hosted on UiPath Automation Cloud. In the cases of Orchestrator 2018.4 and Orchestrator 2019.4, it is necessary to have Organization Units[1] enabled, even if only one Organization Unit is defined (usually named *Default*).

To use Orchestrator Manager, execute it from UiPath Studio by running the **Main.xaml** file in Normal mode[2]. Alternatively, it can also be published as a package to Orchestrator to be deployed to Attended Robots. Since Orchestrator Manager requires user interaction, unattended execution is not possible.

Once the process starts executing, the user is prompted to enter information to log into an Orchestrator instance (Figure 1). More information about this authentication step can be found in section Authentication.

If the authentication is successful, Orchestrator Manager's main control panel (Figure 2) is shown and the user can select an Orchestrator entity and an operation to perform (e.g., Asset creation, Machine deletion, User retrieval and Folder migration).

After confirming the entity and the operation, Orchestrator Manager opens the Excel workbook corresponding to the selected entity and updates it as it performs the selected operation. This workbook can also be used to input data in sheets that create, edit and delete entities.

In cases of operations other than **Get**, each row is processes independently, and the outcome of the operation is written to the column *Result*. If the processing of a row fails, it is necessary to clear the corresponding *Result* column before trying it again, as Orchestrator Manager skips rows that already have a *Result* value specified.

Columns of the workbooks have different colors that represent cells that can be modified by the user (white cells) and cells that are reserved for Orchestrator Manager (gray cells). Refer to section Entities Workbooks for more details about workbooks.

---

[1] From Orchestrator 2019.10, *Organization Units* are referred to as *Classic Folders*.

[2] Execution in Debug mode is not supported.

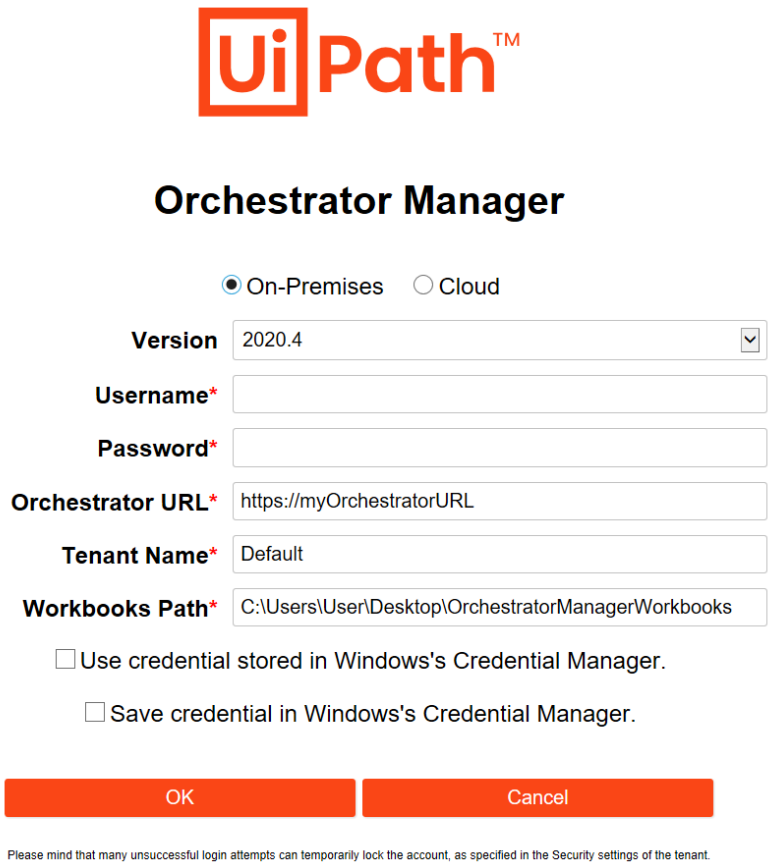



*Figure 1 - Authentication Panel*

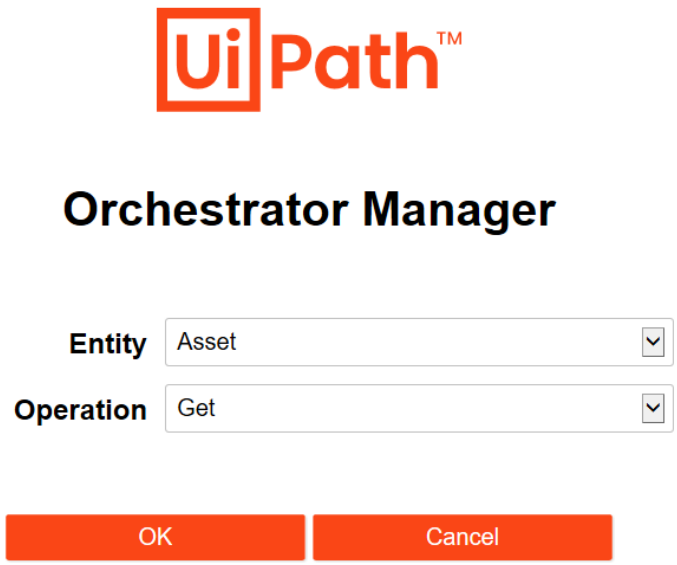



*Figure 2 - Control Panel*

In addition, different operations require different permissions to be set on Orchestrator. For instance, to create an Asset, it is necessary for the user to have the *Assets.Create* permission. For this reason, it is recommended for users of Orchestrator Manager to have the Administrator role, since it usually has all permissions in an Orchestrator tenant. Consult Orchestrator's online documentation for more details about permissions[3].

Other than the Authentication Panel and Control Panel, when it is necessary to choose Folders (Organization Units), Orchestrator Manager shows the Folders (Organization Units) Panel for user selection (Figure 3). Folders[4], formerly known as Organization Units[5], are used to segregate entities into different groups that can be mapped to subdivisions of an organization.

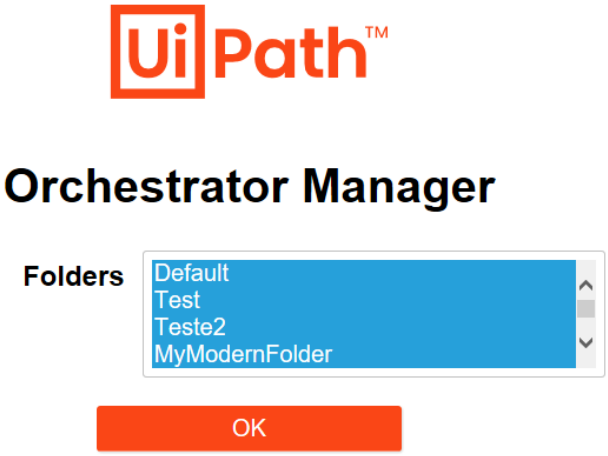



*Figure 3 – Folders (Organization Units) Panel*

## Configuration

The configuration file, **Config.xlsx**, stores settings used throughout Orchestrator Manager and it is divided in three sheets: **Settings**, **Advanced Settings** and **Localization**.

[3] https://docs.uipath.com/orchestrator/reference/permissions-per-endpoint

[4] https://docs.uipath.com/orchestrator/docs/about-folders

[5] https://docs.uipath.com/orchestrator/v2018.4/docs/about-installation

### Settings

The first sheet, **Settings**, contains parameters used to connect to Orchestrator instances. They are divided according to the deployment type of Orchestrator (i.e., on-premises or UiPath Automation Cloud), except *EntitiesWorkbooksFolderPath*, which are used in both cases.

The values of these parameters are used as default values of fields in the Authentication Panel, and they are replaced by new values specified during execution. For this reason, it is not mandatory to make any changes to the **Settings** sheet, as all values can be specified directly in the Authentication Panel.

### Advanced Settings

The **Advanced Settings** sheet contains parameters that usually do not require modification, but that can be changed to adapt Orchestrator Manager to specific situations.

The first parameters can be used to control the throughput of HTTP requests by Orchestrator Manager and reduce its impact on Orchestrator's infrastructure. For example, increasing the interval between requests can be helpful when there is a large number of requests done in a short period.

One especially important parameter from the **Advanced Settings** sheet is *GetCredentialAssetsViaRobot*, which determines whether credential Assets should be retrieved via the robot running Orchestrator Manager instead of using Orchestrator's API. More information about retrieval of credential Assets can be found in Table 1 - Assets.

The **Advanced Settings** sheets also define coordinates of columns in entities workbooks. Changing these usually requires additional modifications to the workflows themselves.

### Localization

The **Localization** sheet contains localization strings that are used in different parts of Orchestrator Manager, such as input forms, paths for workbooks, error messages and warnings.

A new language can be included according to the following steps:

1. Add a new column to the **Localization** sheet and insert the translated strings into the appropriate rows. The header of the new column must be the two-letter representation of the language, according to ISO 639. This header is used by Orchestrator Manager when initializing the *Config* dictionary in the Initialization step.
2. Create a new set of workbooks referring to the new language and place them in a new subfolder of **Workbooks** folder. The name of the new subfolder must be the same two-letter language ISO 639 code used in the first step. The names of sheets of these workbooks must match the names defined in the new language's column in the **Localization** sheet of **Config.xlsx**.
3. Modify the *Switch* activity called *Switch System's Language* in the **Core\InitializeConfigurations.xaml** file by adding a case for the new language. The case *Key* must match the two-letter code that was specified before.

Lastly, some strings have placeholders that are used by the *String.Format()* method, so, although their position may vary, they must be present in the translated string as well.

## Architecture

Figure 4 shows the main steps of Orchestrator Manager, which are detailed in the following subsections.

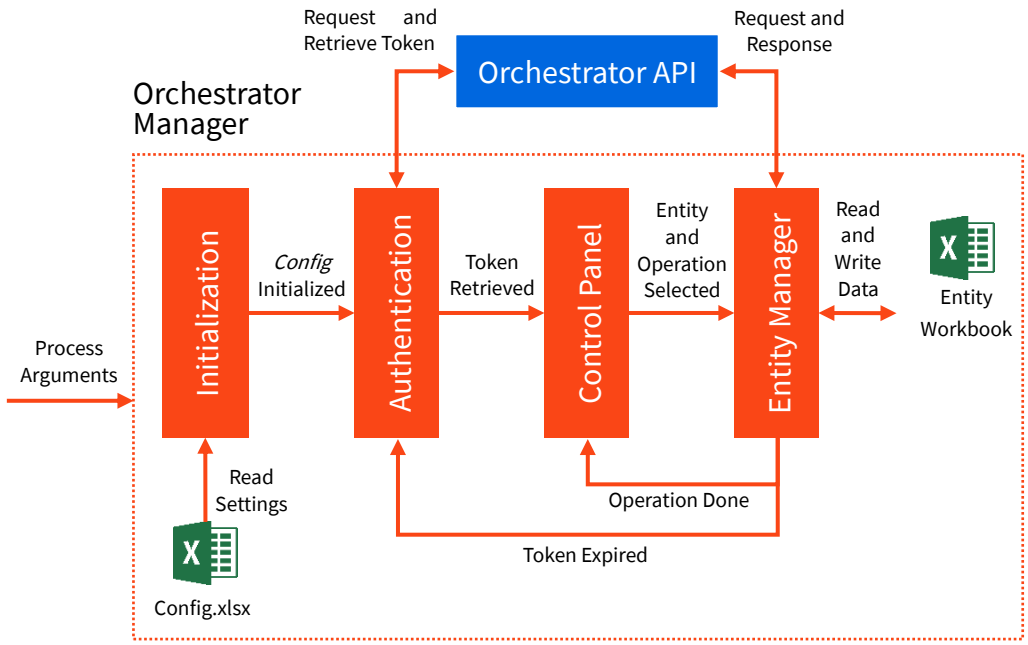



*Figure 4 - Orchestrator Manager Architecture*

### Process Arguments

Orchestrator Manager's entry point, **Main.xaml**, accepts two arguments: *in_Language* and *in_ConfigFilePath*.

The only mandatory argument is *in_ConfigFilePath*, which determines the path to the configuration file.

*in_Language* can be used to define the language to be used by Orchestrator Manager. It must be a two-letter language code according to ISO 639 of a supported language. For more details about localization of Orchestrator Manager, refer to section Localization.

### Initialization

The Initialization step reads process arguments and settings from **Config.xlsx**, storing them in a *Dictionary(Of String, GenericValue)* object called *Config*, which is used throughout Orchestrator Manager's execution.

*Config* also stores localization strings, selected according to the *in_Language* argument or, if this argument is not specified, according to the system's locale settings.

## Authentication

The next step, Authentication, uses information about an Orchestrator instance and authentication credentials to perform an authentication request to Orchestrator's API. This request retrieves an authentication token that is necessary for all other subsequent requests.

The data necessary for authentication depends on whether Orchestrator Manager connects to an instance on-premises or an instance hosted on UiPath Automation Cloud (Figure 1). Refer to the UiPath Automation Cloud online documentation for instructions on how to obtain the parameters Client ID and User Key[6].

Other than the authentication parameters, it is possible to set the path to a local folder to which entity workbooks are copied. If the specified local folder does not exist, Orchestrator Manager automatically creates it before copying the files. By default, the folder is created on the user's Desktop folder.

The Authentication Panel is shown until a token is successfully retrieved or until the user interrupts the execution by pressing the Cancel button or by closing the panel window. Note that multiple unsuccessful login attempts can cause an account lockout according to the tenant's security settings. For more information about account lockout, refer to Orchestrator's documentation[7].

## Control Panel

Once the authentication is successfully done, the user is prompted to choose an entity and an operation to be performed on it. The supported entities are Assets, Environments, Machines, Processes, Users, Robots, Organization Units (only for Orchestrator versions 2018.4 and 2019.4), Folders (for Orchestrator 2019.10 or newer), Queues and Packages. The available operations depend on the entity and are listed in section Entities Workbooks.

Once the user confirms the selection, the appropriate entity manager is invoked based on the chosen entity. After the operation is performed, the execution flow goes back to

---

[6] https://docs.uipath.com/orchestrator/reference/consuming-cloud-api

[7] https://docs.uipath.com/orchestrator/docs/field-descriptions-tenant-settings#section-account-lockout

the Control Panel for the user to choose another operation. This cycle is repeated until the user stops the execution of Orchestrator Manager by clicking on the Cancel button or by closing the Control Panel window (Figure 2). When that happens, Orchestrator Manager automatically closes all entities workbooks opened during its execution.

## Entity Manager

The entity manager of a given entity invokes the workflow that implements the selected operation. This workflow contains all actions necessary to complete the operation, including communication with Orchestrator's API and data input and output using entities workbooks.

If a request fails due to an expired authentication token, Orchestrator Manager prompts the user for credentials to perform the authentication step again for security reasons. After that, it retries the failed request and resumes its regular execution.

## Entities Workbooks

Operations make use of entities workbooks, which are Excel workbooks used to input and output data related to the chosen entities. These workbooks are automatically open by Orchestrator Manager when the corresponding entity is selected in the Control Panel.

Each sheet of an entity workbook represents an operation that can be performed on that entity. As shown in Figure 5, cells of each sheet have different colors that indicate their use:

- White cells can receive input from the user, such as names, types values of Assets to be created.
- Gray cells are used by Orchestrator Manager to write data retrieved from Orchestrator, such as IDs of created Assets. Data input into gray columns is overwritten by Orchestrator Manager.

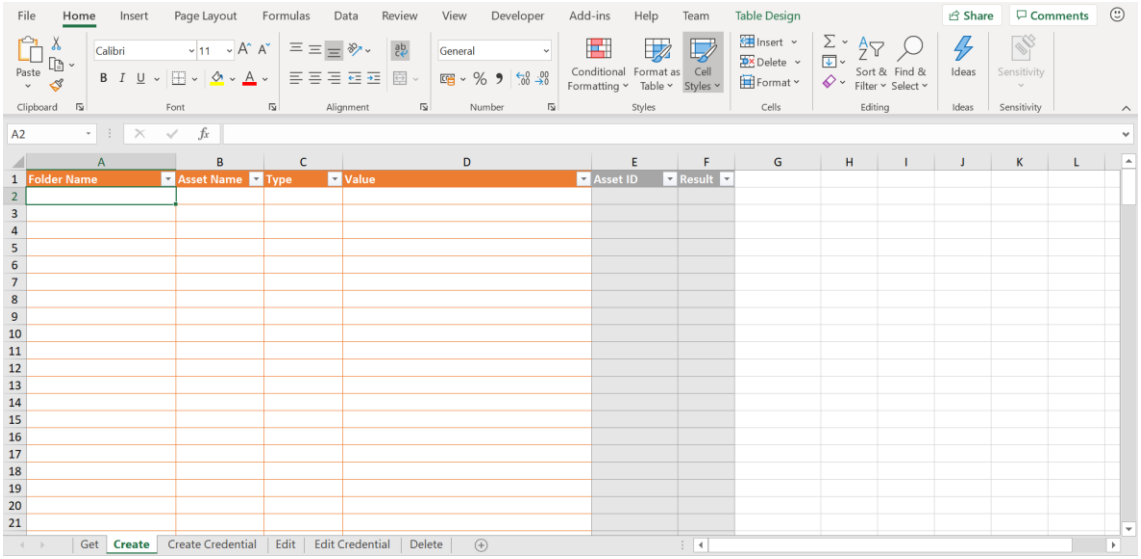



*Figure 5 - Entity Workbook*

The following tables provide more details about workbooks representing each entity: Table 1 - Assets, Table 2 - Environments, Table 3 - Machines, Table 4 - Processes, Table 5 - Robots, Table 6 - Users, Table 7 - Organization Units, Table 8 - Queues, Table 9 - Packages, Table 10 - Folders.

Although some workbooks refer to Folder ID or Folder Name, if Orchestrator's version is 2018.4 or 2019.4, the operations are done using Organization Units.

In addition, some operations are only allowed in the case of Classic Folders (Organization Units), such as the manipulation of Robots and Environments. Similarly, due to its unique User management system, the manipulation of Users is not available when connecting to UiPath Automation Cloud.

When specifying names of subfolders of Modern Folders, it is necessary to input the complete name of the parent Folder, including its ancestors, separated by the forward slash character (*/*). For example, if a Modern Folder called *WorkGroup1* is a subfolder of a Modern Folder called *DepartmentA*, then the complete name of *WorkGroup1* is *DepartmentA/WorkGroup1*. It possible to retrieve the complete names of Folders by performing the **Get** operation on the Folder entity.

Moreover, note that Orchestrator Manager also considers Excel table filters applied to operation sheets. For example, if data about Assets are input into the Create sheet of **Assets.xlsx** and then the table is filtered to show only *Text* Assets, Orchestrator Manager makes requests to create only *Text* Assets.

*Table 1 - Assets*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Assets.<br><br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br><br>For security reasons, passwords of credential Assets are not retrieved by default. However, it is possible to retrieve passwords if the parameter *GetCredentialAssetsViaRobot* in the **Advanced Settings** sheet of the **Config.xlsx** file is set to *TRUE*. This option enables the retrieval of credential Assets via the robot executing Orchestrator Manager, instead of directly from Orchestrator's API, and it has the following restrictions:<br><br>1. The robot executing Orchestrator Manager needs to be connected to Orchestrator.<br>2. If it is a Classic Folder robot, then it can only see credentials Assets in its own Folder. If it is a Modern Folder robot, then it can retrieve credential Assets from other Folders, as long as it has access to them.<br><br>Values of credential Assets (i.e., usernames and passwords) are included in the *Value* field and separated into two lines. |
| **Create** | Creates Assets with the provided data.<br><br>The field *Type* has a fixed number of possible values, according to Asset types available in Orchestrator: *Text*, *Bool* and *Integer*.<br><br>Assets per Robot are not supported. |
| **Create Credential** | Creates credential Assets with the provided data.<br><br>Although credentials are also classified as Assets, they have two values instead of one: *Username* and *Password*.<br><br>Assets per Robot are not supported. |

| | |
|---|---|
| **Edit** | Edits Assets using the provided data.<br><br>It is necessary to specify *Folder Name* and *Asset ID* of the Asset to be edited, and both can be retrieved by the **Get** operation on the Asset entity. The other fields are optional, and fields left blank are not modified.<br><br>It is not possible to change the type of the Asset, and the new value must be compatible with the current type. |
| **Edit Credential** | Edits credential Assets with the provided data.<br><br>It is necessary to specify *Folder Name* and *Asset ID* of the credential Asset to be edited, and both can be retrieved by the **Get** operation on the Asset entity. The other fields are optional, and fields left blank are not modified.<br><br>If *Username* is to be updated, a new password must also be provided. On the other hand, it is possible to update only *Password* by not inputting a new *Username*. |
| **Delete** | Deletes the specified Assets.<br><br>To prevent accidental deletion, it is necessary to provide both ID and name of the Folder containing the Asset, as well as ID and name of each Asset to be deleted. This data can be retrieved by the **Get** operation on the Asset entity. |

*Table 2 - Environments*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Environments.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br>Environments are only available in Classic Folders[8]. |
| **Create** | Creates Environments with the provided data. |
| **Delete** | Deletes the specified Environments.<br>To prevent accidental deletion, it is necessary to provide both ID and name of the Classic Folder containing the Environment, as well as ID and name of each Environment to be deleted. This data can be retrieved by the **Get** operation on the Environment entity. |
| **Add or Remove Robots** | Adds Robots to or remove Robots from the specified Environment.<br>Names of Robots must be provided as comma-separated values (e.g., *Robot1, Robot2, Robot3*). |

*Table 3 - Machines*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Machines.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets. |

[8] https://docs.uipath.com/orchestrator/docs/about-environments

| | |
|---|---|
| **Create** | Creates Machines with the provided data.<br>The field *Type* has a fixed number of possible values, according to Machine types available in Orchestrator: *Standard* and *Template*.<br>A Machine key is automatically generated when a machine is created, and it is written to the **Create** sheet by Orchestrator Manager. |
| **Delete** | Deletes the specified Machines.<br>To prevent accidental deletion, it is necessary to provide both ID and name of each Machine to be deleted. This data can be retrieved by the **Get** operation on the Machine entity. |

*Table 4 - Processes*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Processes.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets. |
| **Create** | Creates Processes with the provided data.<br>*Environment Name* should not be specified for Processes in Modern Folders.<br>*Package Name* and *Package Version* can be retrieved from the Packages page in Orchestrator's web interface or by the **Get** operation on the Package entity. |
| **Delete** | Deletes the specified Processes.<br>To prevent accidental deletion, it is necessary to provide both ID and name of each Process to be deleted. This data can be retrieved by the **Get** operation on the Process entity. |

| | |
|---|---|
| **Update to Latest Package** | Updates the specified Process to use the latest version of the Package. |
| **Rollback to Previous Package** | Updates the specified Process to use the previously used version of the Package. |

*Table 5 - Robots*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Robots.<br><br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br><br>Robots can only be retrieved from Classic Folders. In the case of Modern Folders, Robots are associated with Users and do not represent independent entities. |
| **Create** | Creates Robots with the provided data.<br><br>The field *Hosting Type* has a fixed number of possible values, according to hosting types available in Orchestrator: *Standard* and *Floating*.<br><br>The field *Robot Type* also has a fixed number of possible values, but that can change depending on Orchestrator's version. The possible values for each Orchestrator version supported by Orchestrator Manager are:<br>• 2018.4: *NonProduction*, *Attended*, *Unattended* and *Development*<br>• 2019.4: *NonProduction*, *Attended*, *Unattended* and *Development*<br>• 2019.10: *NonProduction*, *Attended*, *Unattended*, *Development* and *StudioX* |

| | |
|---|---|
| | • 2020.4: *NonProduction*, *Attended*, *Unattended*, *Studio*, *Development*, *StudioX*, *Headless*, *StudioPro* and *TestAutomation*<br>• UiPath Automation Cloud as of September 2020: *NonProduction*, *Attended*, *Unattended*, *Studio*, *Development*, *StudioX*, *Headless*, *StudioPro* and *TestAutomation*<br>The possible values for each Orchestrator version can also be confirmed on Orchestrator's Swagger page[9]. |
| **Edit** | Edits Robots using the provided data.<br>It is necessary to specify *Classic Folder Name* and *Robot ID* of the Robot to be edited, and both can be retrieved by the **Get** operation on the Robot entity. The other fields are optional, and fields left blank are not modified.<br>It is not possible to change the *Hosting Type* of the Robot. |
| **Delete** | Deletes the specified Robots.<br>To prevent accidental deletion, it is necessary to provide both ID and name of the Classic Folder containing the Robot, as well as ID and name of each Robot to be deleted. This data can be retrieved by the **Get** operation on the Robot entity. |

*Table 6 - Users*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Users.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets. |

[9] https://docs.uipath.com/orchestrator/reference/api-references

| | |
|---|---|
| | Users of type Robot are not obtained by this operation, but they can be fetched via the **Get** operation on the Robot entity.<br><br>Although Organization Units to which a User belongs are retrieved by this operation, information about assignment of Users to Folders must be retrieved via the **Get Folders' Users** operations on the Folder entity.<br><br>This operation is not available when connecting to UiPath Automation Cloud. |
| **Create** | Creates Users with the provided data.<br><br>Names of Organization Units and of Roles must be provided as comma-separated values (e.g., *Default, Finance, HR* and *Administrator, Queue Watcher, Robot Creator*). Additionally, it is only possible to specify Organization Units for Users when using on-premises Orchestrator versions 2018.4 and 2019.4.<br><br>The specification of Folders when creating Users is not supported by this operation and should be done via the **Assign User to Folder** operation on the Folder entity.<br><br>The field *Password* must follow the security rules for passwords defined in the tenant's settings page on Orchestrator.<br><br>This operation is not available when connecting to UiPath Automation Cloud. |
| **Edit** | Edits Users using the provided data.<br><br>It is necessary to specify the *ID* of the User to be edited, which can be retrieved by the **Get** operation on the User entity. The other fields are optional, and fields left blank are not modified.<br><br>It is not possible to change the *Username* of the User.<br><br>The field *Status* has a fixed number of possible values, according to hosting types available in Orchestrator: *Active* and *Inactive*. |

| | |
|---|---|
| | For changes regarding User assignment to Organization Units and User Roles at tenant level, refer to the operations **Add or Remove OUs** and **Add or Remove Roles**.<br><br>For changes regarding User assignment to Folders and User Roles at Folder level, refer to operations of the Folder entity.<br><br>This operation is not available when connecting to UiPath Automation Cloud. |
| **Delete** | Deletes the specified Users.<br><br>To prevent accidental deletion, it is necessary to provide both ID and username of each User to be deleted. This data can be retrieved by the **Get** operation on the User entity.<br><br>This operation is not available when connecting to UiPath Automation Cloud. |
| **Add or Remove Roles** | Adds Roles to or remove Roles from the specified User.<br><br>The ID of a User can be retrieved by the **Get** operation on the User entity.<br><br>Names of Roles must be provided as comma-separated values (e.g., *Administrator, Queue Watcher, Robot Creator*). When specifying multiple Roles, if one of them does not exist, the operation is cancelled, and no Roles are added or removed.<br><br>For changes regarding User Roles at Folder level, refer to the operation **Add or Remove User Roles** on the Folder entity.<br><br>This operation is not available when connecting to UiPath Automation Cloud. |
| **Add or Remove OUs** | Adds the specified User to or removes from the specified Organization Units. |

| | |
|---|---|
| | The ID of a User can be retrieved by the **Get** operation on the User entity.<br><br>Names of the Organization Units must be provided as comma-separated values (e.g., *Default, Finance, HR*). When specifying multiple Organization Units, if one of them does not exist, the operation is cancelled, and no Organization Units are added or removed.<br><br>For changes regarding User assignment to Folder, refer to the operations **Assign User to Folder and Unassign User from Folder** on the Folder entity.<br><br>This operation is not available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |

*Table 7 - Organization Units*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Organization Units.<br><br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br><br>This operation is only available in the case of on-premises Orchestrator instances with versions 2018.4 or 2019.4. |
| **Create** | Creates Organization Units with the provided data.<br><br>This operation is only available in the case of on-premises Orchestrator instances with versions 2018.4 or 2019.4. |
| **Delete** | Deletes the specified Organization Units.<br><br>To prevent accidental deletion, it is necessary to provide both ID and name of each Organization Unit to be deleted. This data can be retrieved by the **Get** operation on the Organization Unit entity. |

| | |
|---|---|
| | This operation is only available in the case of on-premises Orchestrator instances with versions 2018.4 or 2019.4. |

*Table 8 - Queues*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Queues.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br>Only data about Queues' definitions is retrieved. Use the **Download Queue Items** operation to retrieve Queue items. |
| **Create** | Creates Queues with the provided data.<br>The fields *Unique Reference* and *Auto Retry* are mandatory and can have two values: *Yes* or *No*. If the field *Auto Retry* has the value *Yes*, the maximum number of retries must be specified as a non-negative integer via the field *Max # of Retries*. |
| **Delete** | Deletes the specified Queues.<br>To prevent accidental deletion, it is necessary to provide both ID and name of the Folder containing the Queue, as well as ID and name of each Queue to be deleted. This data can be retrieved by the **Get** operation on the Queue entity.<br>A Queue can be deleted even if it contains Queue items. |
| **Download Queue Items** | Downloads Queue items from the specified Queue.<br>The field *Download Folder Path* indicates the full path of a local folder to which Queue items should be saved.<br>Items of each Queue is downloaded as a CSV file that follows the same format as the ones downloaded via Orchestrator's web interface. |

| | |
|---|---|
| | Downloaded CSV files are named based on the name of the corresponding Queue and on a timestamp of the moment of download. |
| **Upload Queue Items** | Uploads Queue items to the specified Queue.<br>The values of the field *Queue Items File Path* must be full paths to CSV files containing data about Queue items to be uploaded. The CSV file downloaded via the **Download Queue Items** operation has a particular format that cannot be used by the **Upload Queue Items** operation without additional parsing.<br>The field *Commit Type* has a fixed number of possible values, according to types available in Orchestrator [10]: *AllOrNothing*, *StopOnFirstFailure* and *ProcessAllIndependently*.<br>This operation is not available in the case of on-premises Orchestrator instances with version 2018.4. |

*Table 9 - Packages*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Packages.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br>Each Package version is displayed in a separate row of get **Get** sheet. |
| **Delete** | Deletes the specified Packages.<br>To prevent accidental deletion, it is necessary to provide *ID*, *Version* and *Key* of each Package to be deleted. This data can be retrieved by the **Get** operation on the Package entity. |

[10] https://docs.uipath.com/orchestrator/docs/about-queues-and-transactions#section-upload-strategies

| | |
|---|---|
| **Download** | Downloads the specified Packages.<br><br>The field *Download Folder Path* indicates the full path to a folder to which Packages should be saved. Packages are downloaded as **.nupkg** files, following the same format as the ones downloaded via Orchestrator's web interface.<br><br>Full paths of successfully downloaded Packages are written to the field *Download Package File Path*, and they can be reused by the **Upload** operation.<br><br>Package files are overwritten if they already exist in the specified folder. |
| **Upload** | Uploads the specified Packages.<br><br>The field *Package File Path* indicates the full path to a file of type .nupkg. The name and the version of the Package are automatically defined based on the metadata included in the Package file. |
| **Get Referenced Entities** | Gets Assets and Queues referenced by the specified Package version.<br><br>Orchestrator Manager makes a temporary copy of the specified Package locally and searches for activities that use Assets or Queues in the Package's workflows.<br><br>Details about Assets and Queues used are output to an auxiliary workbook whose path is written to the *Package Referenced Entities File Path* field. |

| | |
|---|---|
| | The Package Entities auxiliary workbook has the following information about referenced Assets and Queues:<br>• Workflow Name: Name of the workflow that references an Asset or to a Queue.<br>• Hierarchy to Activity: Names of activities in the workflow that are ancestors to the activity that references an Asset or a Queue.<br>• Activity Name: Name of the activity that references an Asset (e.g., Get Asset or Get Credential) or a Queue (e.g., Add Queue Item or Get Transaction Item).<br>• Orchestrator Folder Path: Value of the *Orchestrator Folder Path* property of the activity that references an Asset or a Queue. Depending on the versions of activities used by the Package, this property might not be available.<br>• Asset Name (Queue Name): Value of the *AssetName* or *QueueName* property of the activity that references an Asset or a Queue. Variable values are also output. If the Package is based on REFramework [11] and uses a **Config.xlsx** file to store references to Assets and Queues, this file is also checked for the actual name of entities. |

*Table 10 - Folders*

| Operation | Details |
|---|---|
| **Get** | Retrieves data about the existing Folders.<br>This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br>Note that Classic Folders or first level Modern Folders do not have parent Folders. |

[11] https://docs.uipath.com/studio/docs/robotic-enterprise-framework

| | |
|---|---|
| | This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Create** | Creates Folders with the provided data.<br><br>If the Folder to be crated is a subfolder of a Modern Folder, the names of its ancestors should not be included in the *Folder Name* field. Instead, input the complete name of its parent Modern Folder to the *Parent Folder Name* field, including the parent's ancestors if applicable.<br><br>It is not possible to specify parents for Classic Folders.<br><br>The field *Type* has a fixed number of possible values, according to Folder types available in Orchestrator: *Classic* and *Modern*.<br><br>The field *Role Assignment Model* also has a fixed number of possible values, according to the options available in Orchestrator: *Custom Roles* and *Inherit from tenant*. Note that this field only applies for Modern Folders and that a Modern subfolder cannot have a model different than its parent.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Delete** | Deletes the specified Folders.<br><br>To prevent accidental deletion, it is necessary to provide both ID and name of each Folder to be deleted. This data can be retrieved by the **Get** operation on the Folder entity.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Get Folders' Users** | Retrieves data about membership of Users in Folders. |

| | |
|---|---|
| | This sheet is populated by Orchestrator Manager, and the user is not required to input any data. The retrieved data can be copied for use in other sheets.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Assign User to Folder** | Assigns the specified User to the specified Folder.<br><br>The field *User Type* has a fixed number of possible values, according to the types of Users available in Orchestrator: *User, DirectoryUser* and *Directory Group*. Users of type *Robot* are not supported. It possible to retrieve the type and the username of a User by performing the **Get** operation on the User entity.<br><br>The field *Roles Names* must be used in the case of Modern Folders that have *Custom Roles* as Role Assignment Model. It is not possible to specify Roles to Modern Folders if the Role Assignment Model is *Inherit from tenant*.<br><br>In addition, names of Roles must be provided as comma-separated values (e.g., *Administrator, Queue Watcher, Robot Creator*). When specifying multiple Roles, if one of them does not exist, the operation is cancelled, and the User is not assigned to the Folder.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Unassign User from Folder** | Unassigns the specified User from the specified Folder.<br><br>The field *User Type* has a fixed number of possible values, according to the types of Users available in Orchestrator: *User, DirectoryUser* and *DirectoryGroup*. Users of type *Robot* to Folders are not supported. It possible to retrieve the type and the username of a User by performing the **Get** operation on the User entity. |

| | |
|---|---|
| | This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Add or Remove User Roles** | Adds or removes User Roles at Folder level.<br><br>This operation is only supported by Modern Folders.<br><br>The field *User Type* has a fixed number of possible values, according to the types of Users available in Orchestrator: *User, DirectoryUser* and *Directory Group*. Users of type *Robot* are not supported. It possible to retrieve the type and the username of a User by performing the **Get** operation on the User entity.<br><br>Names of Roles in the fields *Names of Roles to Add* and *Names of Roles to Remove* must be provided as comma-separated values (e.g., *Administrator, Queue Watcher, Robot Creator*). When specifying multiple Roles, if one of them does not exist, the operation is cancelled, and no Roles are added or removed. It is not possible to change Roles at Folder level if the Role Assignment Model is *Inherit from tenant*.<br><br>For changes regarding User Roles at tenant level, refer to the operation **Add or Remove Roles** on the User entity.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer. |
| **Migrate Classic to Modern** | Migrates the specified Classic Folder according to the selected migration policies.<br><br>The field *Environment Migration Policy* has a fixed number of values: *Environments to Same Modern Folder* and *Environments to Different Modern Folders*.<br><br>If *Environments to Same Modern Folder* is selected, all Environments in the specified Classic Folder are migrated into a single Modern Folder. In this case, it is possible to specify policies for bulk migration of other entities:<br><br>• Process Migration Policy: Determines whether Processes |

| | |
|---|---|
| | in the Classic Folder should be created in the Modern Folder, according to the following possible values:<br>○ *Migrate All Processes*: Copies all Processes to the target Modern Folder. Modern Folders require that Packages used to create Processes use UiPath.System.Activities version 19.10.1 or newer, and Processes using Packages with older dependencies are not copied and a warning is output.<br>○ *Do Not Migrate Processes*: No Processes are copied to the target Modern Folder.<br>• Asset Migration Policy: Determines how Assets should be created in the Modern Folder, according to the following possible values:<br>○ *Migrate All Assets*: Copies all Assets to the target Modern Folder, including credential Assets. This option requires that the parameter *GetCredentialAssetsViaRobot* in the **Advanced Settings** sheet of the **Config.xlsx** file is set to *TRUE*. Moreover, note that if the migration process is executed by a Robot in a Classic Folder, the Robot cannot access credential Assets in Folders other than the Classic Folder to which it belongs. The migration of Assets per Robot is currently not supported.<br>○ *Migrate Non-Credential Assets*: Only copies Assets of type Text, Integer and Bool. The migration of Assets per Robot is currently not supported.<br>○ *Do Not Migrate Assets*: No Assets are copied to the target Modern Folder.<br>• Queue Migration Policy: Determines how Queues and Queue items should be created in the Modern Folder, |

| | |
|---|---|
| | according to the following possible values:<br>○ *Migrate Queue Definitions and New Items*: Copies all data about Queues, including Queue items with status *New* (migration of Queue items with other statuses is not supported). Queues that have Processes associated with them via the SLA Predictions feature must be copied after the corresponding Process is available in the target Modern Folder. Queues with a large number of items might take a long time to be migrated.<br>○ *Migrate Queue Definitions But Not Queue Items*: Only copies data about Queue definitions, and does not copy Queue items. Queues that have Processes associated with them, either via SLA Predictions or Queue Triggers, must be copied after the corresponding Process is available in the target Modern Folder.<br>○ *Do Not Migrate Queues*: No Queues are copied to the target Modern Folder.<br>• Trigger Migration Policy: Determines whether Triggers should be created in the Modern Folder, according to the following possible values:<br>○ *Migrate All Triggers*: Copies all Triggers to the target Modern Folder. Since Queue Triggers have Processes associated with them, they must be copied after the corresponding Process is available in the target Modern Folder. In addition, migrated Time Triggers utilize Dynamic Allocation as their Execution Target option.<br>○ *Do Not Migrate Triggers*: No Triggers are copied to the target Modern Folder.<br>• Robot Migration Policy: Determines how Robots' settings should be copied to the Modern Folder. Robots in Modern Folders are associated with Users and do not |

| | |
|---|---|
| | represent an independent entity. For this reason, before migrating Robots, it is necessary to manually create Users corresponding to Robots and enable Robot provisioning at User level. Furthermore, the provisioning of Robot at User level must use the same username as the Robot being migrated from the Classic Folder. If Orchestrator Manager can successfully match the username of a Robot provisioned at User level to the username used by a Robot being migrated from the Classic Folder, it proceeds with the migration according to the following possible policies:<br>○ *Overwrite Execution Settings*: Replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder and assigns the corresponding User to the target Modern Folder. The Robot in the Classic Folder remains active.<br>○ *Deactivate Classic Robots*: Assigns the corresponding User to the target Modern Folder and deactivates the Robot in the Classic Folder. Execution settings are not overwritten.<br>○ *Overwrite Execution Settings and Deactivate Classic Robots*: Replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder, assigns the corresponding User to the target Modern Folder and deactivates the Robot in the Classic Folder.<br>○ *Do Not Overwrite Execution Settings Nor Deactivate Classic Robots*: The User corresponding to the Robot to be migrated is assigned to the target Modern Folder, but the execution settings of the User's associated Robot is not replaced and the Robot in the Classic Folder |

| | |
|---|---|
| | is not deactivated.<br>○ *Do Not Migrate Robots*: No Users are assigned to the target Modern Folder, execution settings are not overwritten, and Robots in the Classic Folder remain active.<br><br>Regarding migration of Robots, note that if an Attended or Studio Robot is defined at User level and active in a Classic Folder at the same time, the connection to the one in the Classic Folder takes precedence.<br><br>Finer adjustments related to Users and Folders, such as the management of Folder Roles, can be done via Orchestrator's web interface or using operations of the Folder entity.<br><br>Error or warnings that occur during the migration using the option *Environments to Same Modern Folder* are output to a log file, and the path to this log file is written to the *Result* field.<br><br>If *Environments to Different Modern Folders* is selected, it is possible to specify details about migration of individual entities. In this case, it is not necessary to specify values for the remaining fields of the **Migrate Classic to Modern** sheet, as a new migration workbook for the specified Folder is created and shown to the user. Details of migration workbooks can be found in Table 11.<br><br>This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2020.10 or newer. |

*Table 11 – Migration from Classic to Modern*

| Entity | Details |
|---|---|
| **Processes** | Specifies what Processes should be copied to what Modern Folders.<br><br>Modern Folders require that Packages used to create Processes use UiPath.System.Activities version 19.10.1 or newer, and Processes using Packages with older dependencies are not copied and a warning is output.<br><br>If a Process needs to be copied to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Process and specify different target Modern Folders. |
| **Assets** | Specifies what Assets should be copied to what Modern Folders.<br><br>The migration of credential Assets requires that the parameter *GetCredentialAssetsViaRobot* in the **Advanced Settings** sheet of the **Config.xlsx** file is set to *TRUE*. In addition, note that if the migration process is executed by a Robot in a Classic Folder, the Robot cannot access credential Assets in Folders other than the Classic Folder to which it belongs. The migration of Assets per Robot is currently not supported.<br><br>If an Asset needs to be copied to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Asset and specify different target Modern Folders.<br><br>The migration of Assets per Robot is currently not supported. |
| **Queues** | Specifies what Queues should be copied to what Modern Folders.<br><br>It is possible to specify the migration policy, according to the following values:<br><br>• *Migrate Queue Definition and New Items*: Copies all data about the Queue, including Queue items with status *New* (migration of Queue items with other statuses is not supported). Queues with a large number of items might |

| | |
|---|---|
| | take a long time to be migrated.<br>• *Migrate Queue Definition But Not Queue Items*: Only copies data about the Queue definition, and does not copy Queue items.<br>Queues that have Processes associated with them via the SLA Predictions feature must be copied after the corresponding Process is available in the target Modern Folder.<br>If a Queue needs to be copied to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Queue and specify different target Modern Folders. |
| **Triggers** | Specifies what Triggers should be copied to what Modern Folders.<br>Since Queue Triggers have Processes associated with them, they must be copied after the corresponding Process is available in the target Modern Folder.<br>Migrated Time Triggers utilize Dynamic Allocation as their Execution Target option.<br>If a Trigger needs to be copied to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Trigger and specify different target Modern Folders. |
| **Robots** | Specifies what Robots' settings should be copied to what Modern Folders.<br>Robots in Modern Folders are associated with Users and do not represent an independent entity. For this reason, before migrating Robots, it is necessary to create Users corresponding to Robots and enable Robot provisioning at User level. Moreover, the provisioning of Robot at User level must use the same username as the Robot being migrated from the Classic Folder.<br>If Orchestrator Manager can successfully match the username of a Robot provisioned at User level to the username used by a Robot being migrated from the Classic Folder, it proceeds with the migration according to the following possible policies: |

| | |
|---|---|
| | • *Overwrite Execution Settings*: Replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder and assigns the corresponding User to the target Modern. The Robot in the Classic Folder remains active.<br>• *Deactivate Classic Robots*: Assigns the corresponding User to the target Modern Folder and deactivates the Robot in the Classic Folder. Execution settings are not overwritten.<br>• *Overwrite Execution Settings and Deactivate Classic Robot*: Replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder, assigns the corresponding User to the target Modern Folder and deactivates the Robot in the Classic Folder.<br>• *Do Not Overwrite Execution Settings Nor Deactivate Classic Robot*: The User corresponding to the Robot to be migrated is assigned to the target Modern Folder, but the execution settings of the User's associated Robot is not replaced and the Robot in the Classic Folder is not deactivated.<br><br>If an Attended or Studio Robot is defined at User level and active in a Classic Folder at the same time, the connection to the one in the Classic Folder takes precedence.<br><br>If a User corresponding to a Robot needs to be assigned to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Robot and specify different target Modern Folders.<br><br>Finer adjustments related to Users and Folders, such as the management of Folder Roles, can be done via Orchestrator's web interface or using operations of the Folder entity. |

# Examples of Use

This section present examples of use cases covered by Orchestrator Manager.

## Bulk Operations on Entities

A simple but common scenario faced by Orchestrator administrators is the definition of a large number of entities. For example, an administrator might be asked to provision 30 Floating Attended Robots, equally distributed among three Environments: Finance, HR and Legal. In addition, Robots from different Environments should use

The first step to do this with Orchestrator Manager is to, after starting its execution, open the **Robots.xlsx** workbook and define details about Robots to be provisioned on the **Create** sheet, as shown in Figure 6. Since all Robots are Floating Attended Robots, there is no need to input Machine name nor password.

After that, use the Control Panel to choose the entity Robot and the operation Create. Orchestrator Manager will create all 30 Robots and update the **Create** sheet with the IDs of the created Robots.

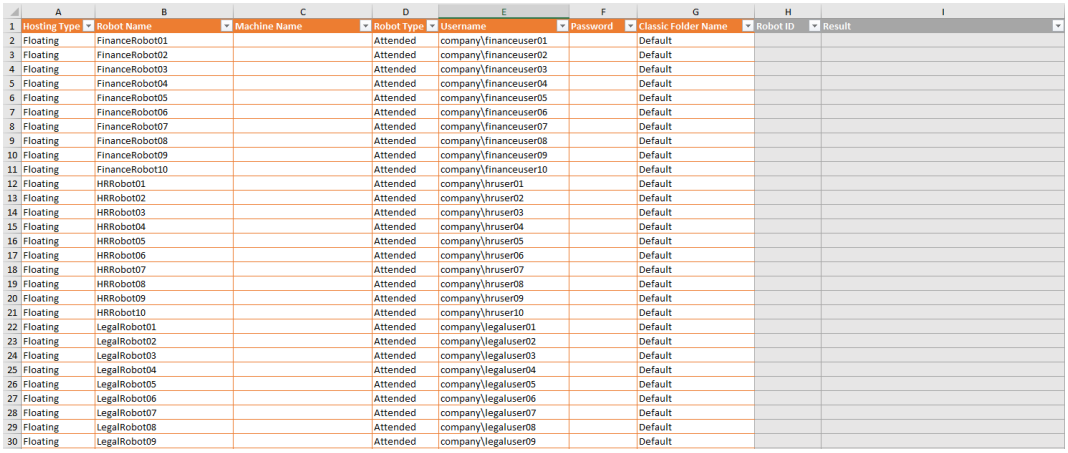
Then, open the **Environments.xlsx** workbook and input details for the creation of three new Environments (Figure 7) in the **Create** sheet, followed by choosing the Environment entity and the Create operation on Orchestrator's Control Panel.

Once Robots and Environments are created, the final step is to link them. To do so, copy the names of the created Robots from the **Robots.xlsx** workbook and paste them in the column *Names of Robots to Add* of the **Add or Remove Robots** sheet in the **Environments.xlsx** workbook. Do the same steps to copy names of Environments and the name of the Classic Folder. Figure 8 shows the filled sheet with one Robot added per row, but it is also possible to define multiple Robots in the same row, as described in Table 2.

After preparing the sheet **Add or Remove Robots**, use the Control Panel to choose the entity Environment and the operation Add or Remove Robots. Once Orchestrator Manager finishes processing, the final result can be confirmed on Orchestrator's web interface (Figure 9).

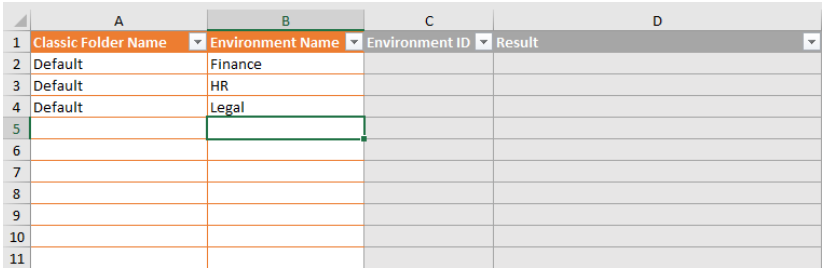
This example highlights how to use Orchestrator Manager to perform the same operation over a large number of entities by specifying details on entities workbooks and confirming operations on the Control Panel.

Note that in the case of a large number of entities to be manipulated, visualizing them in a spreadsheet can help Orchestrator administrator have a more comprehensive view of the task at hand. Also, it is possible to leverage Excel capabilities for faster text editing and creation of incremental values, such as the Auto Fill feature[12].



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Hosting Type | Robot Name | Machine Name | Robot Type | Username | Password | Classic Folder Name | Robot ID | Result |
| 2 | Floating | FinanceRobot01 | | Attended | company\financeuser01 | | Default | | |
| 3 | Floating | FinanceRobot02 | | Attended | company\financeuser02 | | Default | | |
| 4 | Floating | FinanceRobot03 | | Attended | company\financeuser03 | | Default | | |
| 5 | Floating | FinanceRobot04 | | Attended | company\financeuser04 | | Default | | |
| 6 | Floating | FinanceRobot05 | | Attended | company\financeuser05 | | Default | | |
| 7 | Floating | FinanceRobot06 | | Attended | company\financeuser06 | | Default | | |
| 8 | Floating | FinanceRobot07 | | Attended | company\financeuser07 | | Default | | |
| 9 | Floating | FinanceRobot08 | | Attended | company\financeuser08 | | Default | | |
| 10 | Floating | FinanceRobot09 | | Attended | company\financeuser09 | | Default | | |
| 11 | Floating | FinanceRobot10 | | Attended | company\financeuser10 | | Default | | |
| 12 | Floating | HRRobot01 | | Attended | company\hruser01 | | Default | | |
| 13 | Floating | HRRobot02 | | Attended | company\hruser02 | | Default | | |
| 14 | Floating | HRRobot03 | | Attended | company\hruser03 | | Default | | |
| 15 | Floating | HRRobot04 | | Attended | company\hruser04 | | Default | | |
| 16 | Floating | HRRobot05 | | Attended | company\hruser05 | | Default | | |
| 17 | Floating | HRRobot06 | | Attended | company\hruser06 | | Default | | |
| 18 | Floating | HRRobot07 | | Attended | company\hruser07 | | Default | | |
| 19 | Floating | HRRobot08 | | Attended | company\hruser08 | | Default | | |
| 20 | Floating | HRRobot09 | | Attended | company\hruser09 | | Default | | |
| 21 | Floating | HRRobot10 | | Attended | company\hruser10 | | Default | | |
| 22 | Floating | LegalRobot01 | | Attended | company\legaluser01 | | Default | | |
| 23 | Floating | LegalRobot02 | | Attended | company\legaluser02 | | Default | | |
| 24 | Floating | LegalRobot03 | | Attended | company\legaluser03 | | Default | | |
| 25 | Floating | LegalRobot04 | | Attended | company\legaluser04 | | Default | | |
| 26 | Floating | LegalRobot05 | | Attended | company\legaluser05 | | Default | | |
| 27 | Floating | LegalRobot06 | | Attended | company\legaluser06 | | Default | | |
| 28 | Floating | LegalRobot07 | | Attended | company\legaluser07 | | Default | | |
| 29 | Floating | LegalRobot08 | | Attended | company\legaluser08 | | Default | | |
| 30 | Floating | LegalRobot09 | | Attended | company\legaluser09 | | Default | | |

*Figure 6 - Create Robots*



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Classic Folder Name | Environment Name | Environment ID | Result |
| 2 | Default | Finance | | |
| 3 | Default | HR | | |
| 4 | Default | Legal | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |

*Figure 7 - Create Environments*

12 https://support.microsoft.com/en-us/office/fill-data-automatically-in-worksheet-cells-74e31bdd-d993-45da-aa82-35a236c5b5db

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Classic Folder Name | Environment Name | Names of Robots to Add | Names of Robots to Remove | Result |
| 2 | Default | Finance | FinanceRobot01 | | |
| 3 | Default | Finance | FinanceRobot02 | | |
| 4 | Default | Finance | FinanceRobot03 | | |
| 5 | Default | Finance | FinanceRobot04 | | |
| 6 | Default | Finance | FinanceRobot05 | | |
| 7 | Default | Finance | FinanceRobot06 | | |
| 8 | Default | Finance | FinanceRobot07 | | |
| 9 | Default | Finance | FinanceRobot08 | | |
| 10 | Default | Finance | FinanceRobot09 | | |
| 11 | Default | Finance | FinanceRobot10 | | |
| 12 | Default | HR | HRRobot01 | | |
| 13 | Default | HR | HRRobot02 | | |
| 14 | Default | HR | HRRobot03 | | |
| 15 | Default | HR | HRRobot04 | | |
| 16 | Default | HR | HRRobot05 | | |
| 17 | Default | HR | HRRobot06 | | |
| 18 | Default | HR | HRRobot07 | | |
| 19 | Default | HR | HRRobot08 | | |
| 20 | Default | HR | HRRobot09 | | |
| 21 | Default | HR | HRRobot10 | | |
| 22 | Default | Legal | LegalRobot01 | | |
| 23 | Default | Legal | LegalRobot02 | | |
| 24 | Default | Legal | LegalRobot03 | | |
| 25 | Default | Legal | LegalRobot04 | | |
| 26 | Default | Legal | LegalRobot05 | | |
| 27 | Default | Legal | LegalRobot06 | | |
| 28 | Default | Legal | LegalRobot07 | | |
| 29 | Default | Legal | LegalRobot08 | | |
| 30 | Default | Legal | LegalRobot09 | | |

*Figure 8 - Add Robots to Environments*

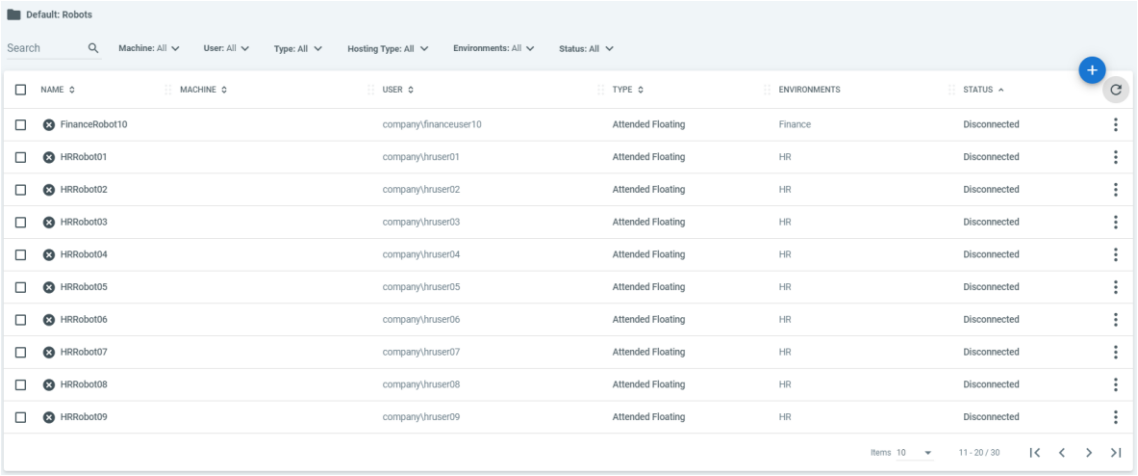



*Figure 9 – Robots Provisioned and Added to Environments*

## Migration of Entities Between Tenants

Another situation faced by Orchestrator administrators is the migration or copying of entities between tenants. Although this is a less common scenario, it can happen when moving an on-premises tenant to a tenant hosted on UiPath Automation Cloud, or when copying entities from a tenant used for development purposes to a tenant used for testing purposes. In both cases, migrations require a large effort by administrators and can also introduce mistakes or inconsistencies due to manually copying details about entities.

To facilitate the migration process, administrators can use Orchestrator Manager's Get and Create operations to export and import entities between tenants. The steps to

complete such procedure are illustrated by the following example:

1. Connect Orchestrator Manager to the source tenant (Figure 10).
2. Use the Get operation of each entity to be copied, retrieving data from the source tenant (Figure 11).
3. For each entity, copy data from the **Get** sheet to the **Create** sheet, modifying values if necessary (Figure 12). The phases of importing and exporting the data can happen at different points in time, providing flexibility for adapting the imported data to requirements of different tenants.
4. Disconnect Orchestrator Manager from the source tenant.
5. Connect Orchestrator Manager to the target tenant (Figure 13).
6. Use the Create operation of each entity to be copied, creating all entities in the target tenant. Note that some entities have to be copied before others. For example, it is necessary to copy Packages before creating Processes that use them. Also, entities that belong to Folders must have their Folders created before being copied.

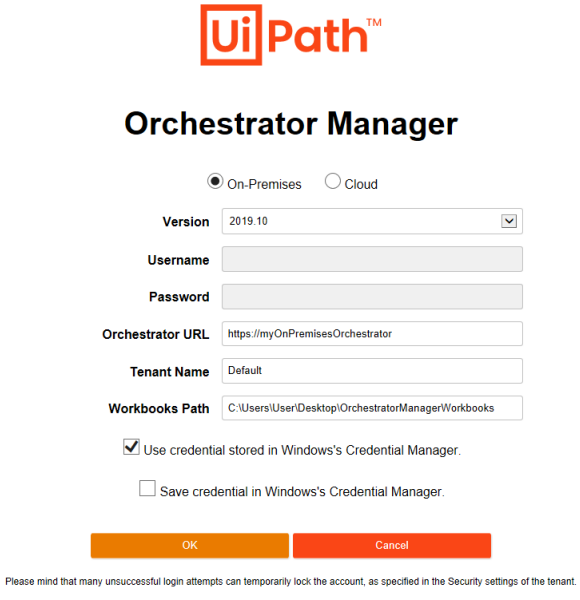



*Figure 10 - Connection to On-Premises Source Tenant*

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Folder ID | Folder Name | Asset ID | Asset Name | Type | Value |
| 2 | 1 | Default | 1 | TextAsset001 | Text | ABC001 |
| 3 | 1 | Default | 2 | TextAsset002 | Text | ABC002 |
| 4 | 1 | Default | 3 | TextAsset003 | Text | ABC003 |
| 5 | 1 | Default | 4 | TextAsset004 | Text | ABC004 |
| 6 | 1 | Default | 5 | TextAsset005 | Text | ABC005 |
| 7 | 1 | Default | 6 | TextAsset006 | Text | ABC006 |
| 8 | 1 | Default | 7 | TextAsset007 | Text | ABC007 |
| 9 | 1 | Default | 8 | TextAsset008 | Text | ABC008 |
| 10 | 1 | Default | 9 | TextAsset009 | Text | ABC009 |
| 11 | 1 | Default | 10 | TextAsset010 | Text | ABC010 |
| 12 | 1 | Default | 11 | TextAsset011 | Text | ABC011 |
| 13 | 1 | Default | 12 | TextAsset012 | Text | ABC012 |
| 14 | 1 | Default | 13 | TextAsset013 | Text | ABC013 |
| 15 | 1 | Default | 14 | TextAsset014 | Text | ABC014 |
| 16 | 1 | Default | 15 | TextAsset015 | Text | ABC015 |
| 17 | 1 | Default | 16 | TextAsset016 | Text | ABC016 |
| 18 | 1 | Default | 17 | TextAsset017 | Text | ABC017 |
| 19 | 1 | Default | 18 | TextAsset018 | Text | ABC018 |
| 20 | 1 | Default | 19 | TextAsset019 | Text | ABC019 |
| 21 | 1 | Default | 20 | BoolAsset001 | Bool | FALSE |
| 22 | 1 | Default | 21 | BoolAsset002 | Bool | FALSE |
| 23 | 1 | Default | 22 | BoolAsset003 | Bool | FALSE |
| 24 | 1 | Default | 23 | BoolAsset004 | Bool | FALSE |
| 25 | 1 | Default | 24 | BoolAsset005 | Bool | FALSE |
| 26 | 1 | Default | 25 | BoolAsset006 | Bool | FALSE |
| 27 | 1 | Default | 26 | BoolAsset007 | Bool | FALSE |
| 28 | 1 | Default | 27 | BoolAsset008 | Bool | FALSE |
| 29 | 1 | Default | 28 | BoolAsset009 | Bool | FALSE |
| 30 | 1 | Default | 29 | BoolAsset010 | Bool | FALSE |
| 31 | 1 | Default | 30 | BoolAsset011 | Bool | FALSE |
| 32 | 1 | Default | 31 | BoolAsset012 | Bool | FALSE |
| 33 | 1 | Default | 32 | BoolAsset013 | Bool | FALSE |
| 34 | 1 | Default | 33 | BoolAsset014 | Bool | FALSE |
| 35 | 1 | Default | 34 | BoolAsset015 | Bool | FALSE |
| 36 | 1 | Default | 35 | BoolAsset016 | Bool | FALSE |
| 37 | 1 | Default | 36 | IntegerAsset001 | Integer | 1001 |
| 38 | 1 | Default | 37 | IntegerAsset002 | Integer | 1002 |

Get | Create | Create Credential | Edit | Edit Credential | Delete

*Figure 11 - Get Assets from Source Tenant*

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Folder Name | Asset Name | Type | Value | Asset ID | Result |
| 2 | ModernFolder01 | TextAsset001 | Text | ABC001 | | |
| 3 | ModernFolder01 | TextAsset002 | Text | ABC002 | | |
| 4 | ModernFolder01 | TextAsset003 | Text | ABC003 | | |
| 5 | ModernFolder01 | TextAsset004 | Text | ABC004 | | |
| 6 | ModernFolder01 | TextAsset005 | Text | ABC005 | | |
| 7 | ModernFolder01 | TextAsset006 | Text | ABC006 | | |
| 8 | ModernFolder01 | TextAsset007 | Text | ABC007 | | |
| 9 | ModernFolder01 | TextAsset008 | Text | ABC008 | | |
| 10 | ModernFolder01 | TextAsset009 | Text | ABC009 | | |
| 11 | ModernFolder01 | TextAsset010 | Text | ABC010 | | |
| 12 | ModernFolder01 | TextAsset011 | Text | ABC011 | | |
| 13 | ModernFolder01 | TextAsset012 | Text | ABC012 | | |
| 14 | ModernFolder01 | TextAsset013 | Text | ABC013 | | |
| 15 | ModernFolder01 | TextAsset014 | Text | ABC014 | | |
| 16 | ModernFolder01 | TextAsset015 | Text | ABC015 | | |
| 17 | ModernFolder01 | TextAsset016 | Text | ABC016 | | |
| 18 | ModernFolder01 | TextAsset017 | Text | ABC017 | | |
| 19 | ModernFolder01 | TextAsset018 | Text | ABC018 | | |
| 20 | ModernFolder01 | TextAsset019 | Text | ABC019 | | |
| 21 | ModernFolder01 | BoolAsset001 | Bool | FALSE | | |
| 22 | ModernFolder01 | BoolAsset002 | Bool | FALSE | | |
| 23 | ModernFolder01 | BoolAsset003 | Bool | FALSE | | |
| 24 | ModernFolder01 | BoolAsset004 | Bool | FALSE | | |
| 25 | ModernFolder01 | BoolAsset005 | Bool | FALSE | | |
| 26 | ModernFolder01 | BoolAsset006 | Bool | FALSE | | |
| 27 | ModernFolder01 | BoolAsset007 | Bool | FALSE | | |
| 28 | ModernFolder01 | BoolAsset008 | Bool | FALSE | | |
| 29 | ModernFolder01 | BoolAsset009 | Bool | FALSE | | |
| 30 | ModernFolder01 | BoolAsset010 | Bool | FALSE | | |
| 31 | ModernFolder01 | BoolAsset011 | Bool | FALSE | | |
| 32 | ModernFolder01 | BoolAsset012 | Bool | FALSE | | |
| 33 | ModernFolder01 | BoolAsset013 | Bool | FALSE | | |
| 34 | ModernFolder01 | BoolAsset014 | Bool | FALSE | | |
| 35 | ModernFolder01 | BoolAsset015 | Bool | FALSE | | |
| 36 | ModernFolder01 | BoolAsset016 | Bool | FALSE | | |
| 37 | ModernFolder01 | IntegerAsset001 | Integer | 1001 | | |
| 38 | ModernFolder01 | IntegerAsset002 | Integer | 1002 | | |

Get | Create | Create Credential | Edit | Edit Credential | Delete

*Figure 12 - Details of Assets to Create*

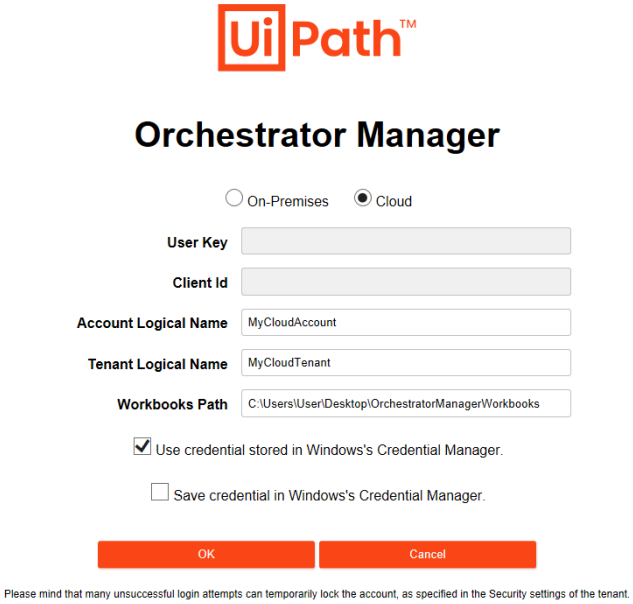



*Figure 13 - Connection to Cloud Target Tenant*

## Migration of Classic Folders to Modern Folders

Another type of migration situation faced by Orchestrator administrators is the one from Classic to Modern Folders. Modern Folders, introduced in Orchestrator 2019.10, offer new features to facilitate the management of large RPA implementations. However, similarly to the migration between tenants, manually copying all entities from a Classic Folder to a Modern Folder can also be very demanding.

Orchestrator Manager includes the operation Migrate Classic to Modern on the Folder entity (Table 10), which can be used to accelerate the migration procedure and reduce the risk of mistakes.

Before starting the migration, a few conditions should be fulfilled:

1. Processes in Modern Folders must use Packages that have the dependency UiPath.System.Activities version 19.10.1 or newer. For this reason, before migrating Processes, it is recommended to republish the corresponding Packages using the latest versions of their dependencies.
2. Modern Folders do not defined Robots as an independent entity. Instead, they are associated with Users at tenant level[13], and those Users should be included

[13] https://docs.uipath.com/orchestrator/docs/managing-users

in the appropriate Folders[14]. Because of this, it is necessary to create setup Users for the migration, adding new Users if necessary and enabling their Robots with the appropriate usernames and Robot settings (Figure 14). Note that for Orchestrator Manager to migrate a Robot from a Classic Folder to a Modern Folder, the username defined for a User's Robot must match the username used by the Robot migrated from the Classic Folder.

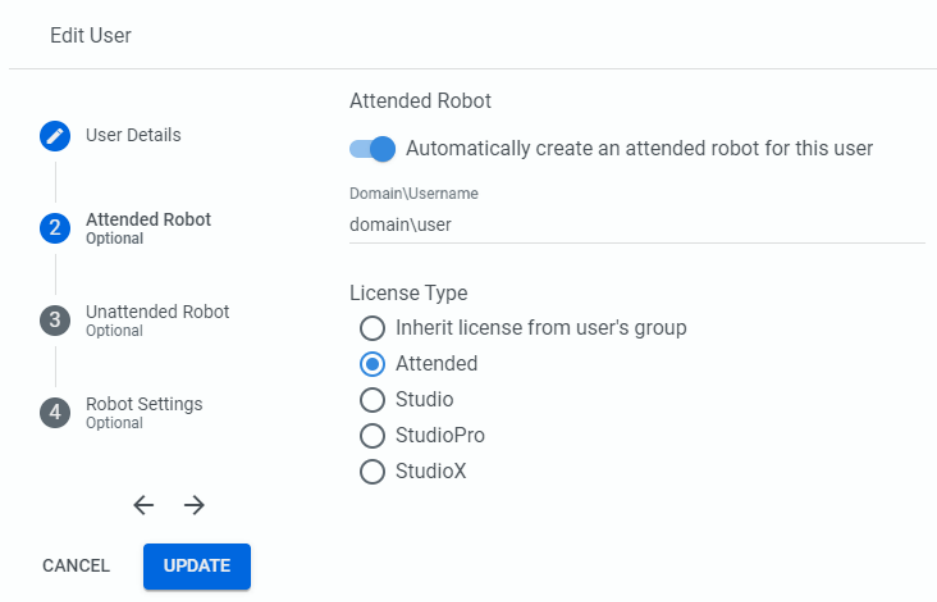



*Figure 14 - Enabling a User's Attended Robot*

In addition, as mentioned in Table 10, an important decision to take when migrating to Modern Folders is what should be done regarding Environments defined in the Classic Folder. There are three common options:

1. All Environments in the Classic Folder are mapped into a single target Modern Folder. This can be a good alternative if there is only a single Environment defined in the Classic Folder or if the Users (and their associated Robots) assigned to the target Modern Folder require access to all entities in the Folder.
2. Each Environment in the Classic Folder is mapped to a different Modern Folder. This is a common option when there needs to be a separation of access rights that was originally based on different Environments. However, since entities like Assets and Queues are also separated in Folders, other than Modern Folders representing Environments, there can also be an additional, auxiliary

[14] https://docs.uipath.com/orchestrator/docs/managing-folders#section-adding-local-users

Modern Folder that stores shared entities.

3. Each Environment in the Classic Folder is mapped into a subfolder of a Modern Folder. This is a more specific variation of the second option above, in which the feature of Modern Folder hierarchy is taken into consideration. This option is suitable, for example, if there needs to be a User with monitoring rights for a parent Modern Folder and its children, but without the ability to run Processes.

The first option can be implemented by inputting the necessary migration policies in the **Migrate Classic to Modern** sheet of the **Folders.xlsx** workbook, as described in Table 11. For the second and third options, Orchestrator Manager makes use of a dedicated migration workbook for each Classic Folder to be migrated (Table 11).

As an example of migration using the second option (i.e., mapping of different Environments to different Modern Folders), consider the structure of a Classic Folder, named *SourceClassic*, shown in Figure 15. *SourceClassic* contains five Assets (*A1*, *A2*, *A3*, *A4*, *A5*), four Floating Attended Robots (*R1*, *R2*, *R3*, *R4*), three Queues (*Q1*, *Q2*, *Q3*), three Time Triggers (*T1*, *T2*, *T3*), three Environments (*E1*, *E2*, *E3*) and five Processes (*P1*, *P2*, *P3*, *P4*, *P5*).

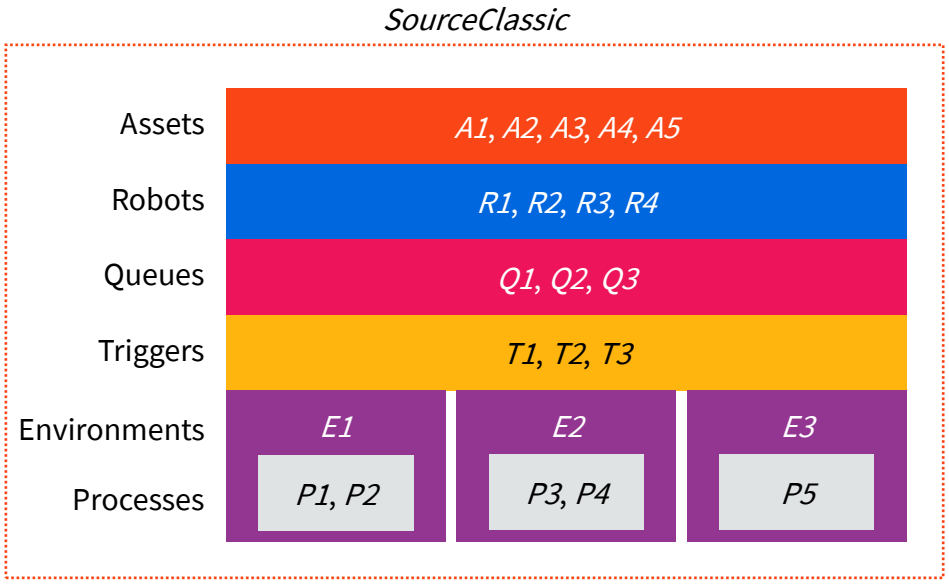



*Figure 15 - Example of Classic Folder* SourceClassic

The relationships between these entities are represented in the following figures. For instance, Figure 16 shows that Process *P1* utilizes Assets *A1* and *A3* as well as Queue *Q1*; Figure 17 depicts Robots *R1* and *R2* assigned Environment *E1*; and Figure 18 illustrates that Time Trigger *T1* triggers Process *P1*.

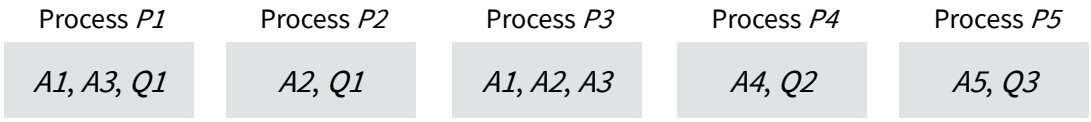



*Figure 16 – Utilization of Assets and Queues by Processes in Classic Folder* SourceClassic


*Figure 17 – Assignment of Robots to Environments in Classic Folder* SourceClassic

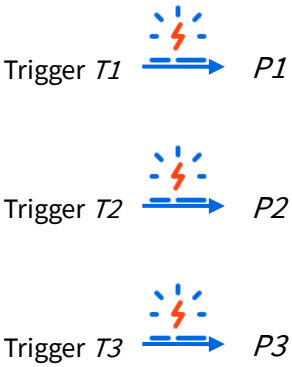



*Figure 18 – Relationship Between Triggers and Triggered Processes in Classic Folder* SourceClassic

Based on such relationships and the decision to have Environments mapped into different Modern Folders, a possible configuration of the equivalent entities using Modern Folders can be seen in Figure 19 and it is detailed as follows:

- It is assumed that Users were created and enabled with Robots for each Robot in the Classic Folder *SourceClassic*, so Robot *R1* is associated with User *U1*, *R2* is associated with *U2* and so on. The usernames of Robots provisioned to *U1*, *U2*, *U3* and *U4* match the usernames used by Robots *R1*, *R2*, *R3* and *R4*.
- For each Process in Classic Folder *SourceClassic*, a corresponding Process is created in a different Modern Folder representing the associated Environment. For instance, since Environment *E1* is mapped into Folder *Modern1*, Processes *P1* and *P2* are created in *Modern1*.
- Assets and Queues are created in the same Folders of Processes that utilize them, such as Asset *A4* created in *Modern2* and Queue *Q3* created in *Modern3*.

However, in this configuration, entities shared by multiple Processes are included in an auxiliary Modern Folder, *ModernAuxiliary*, such as Assets *A1*, *A2*, *A3*, and Queue *Q1*. In a more sophisticated case, if any of the Queues had Triggers associated with them (i.e., Queue Triggers), it would be necessary to place the Queue, the associated Queue Trigger and the Process to be triggered in the same Modern Folder.

- Time Triggers are created in the same Modern Folders of Processes that utilize them. For example, Time Trigger *T3* is created in *Modern2* because this is the Modern Folder that contains Process *P3*. As mentioned before, in the case of Queue Triggers, it is necessary that the Trigger is created in the same Folder as the Queue and as the Process to be triggered.
- Users that represent Robots are added to Folders containing Processes that they should execute and to Folders containing other entities used by those Processes. For instance, Users *U1* and *U2* are added to *Modern1* because, Processes *P1* and *P2* are in *Modern1*. They are also assigned to *ModernAuxiliary* since they need access to Assets and Queues defined in *ModernAuxiliary*: *A1*, *A2*, *A3* and *Q1*. On the other hand, User *U4* can be assigned only to *Modern3* because the Processes it should execute (i.e., *P5*) and entities related to such Processes (i.e., *A5* and *Q3*) are all in *Modern3*.

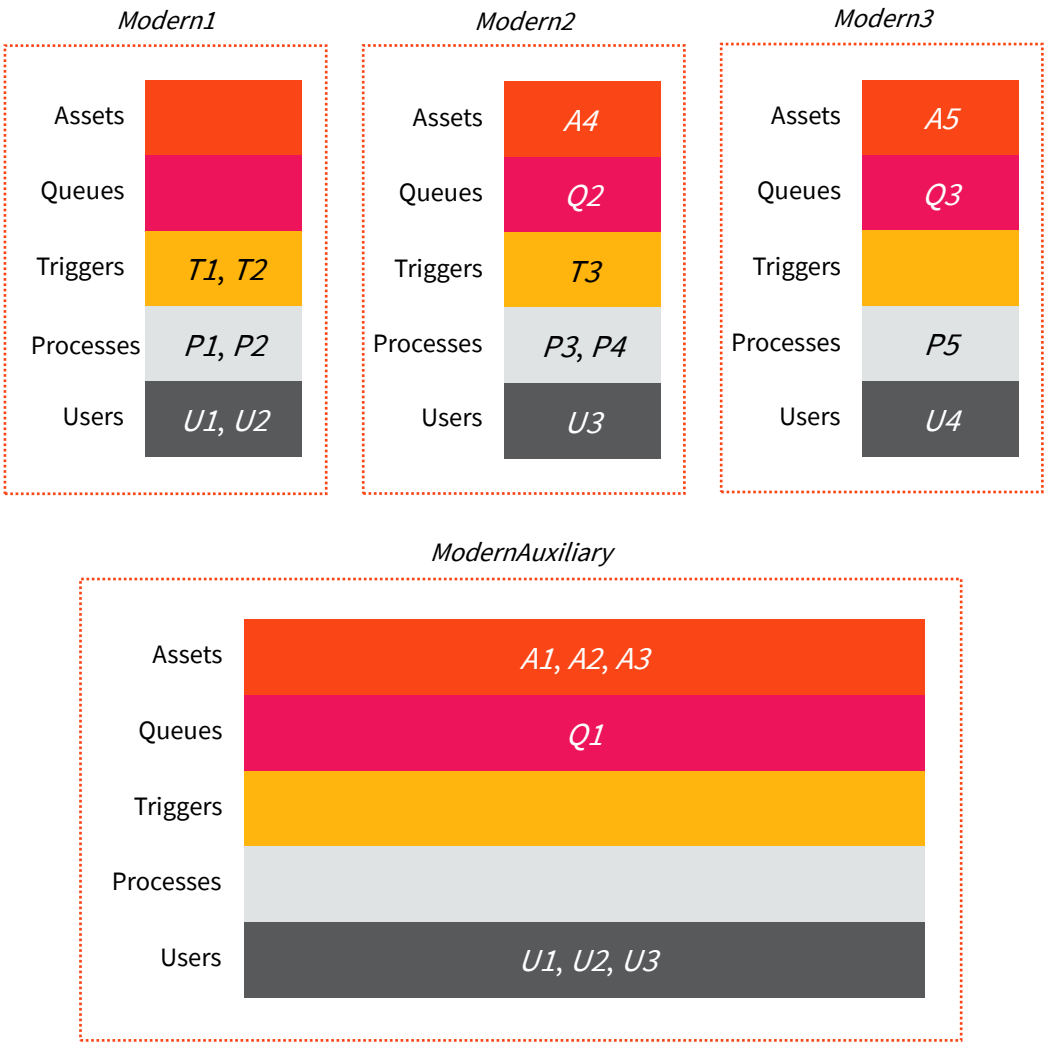
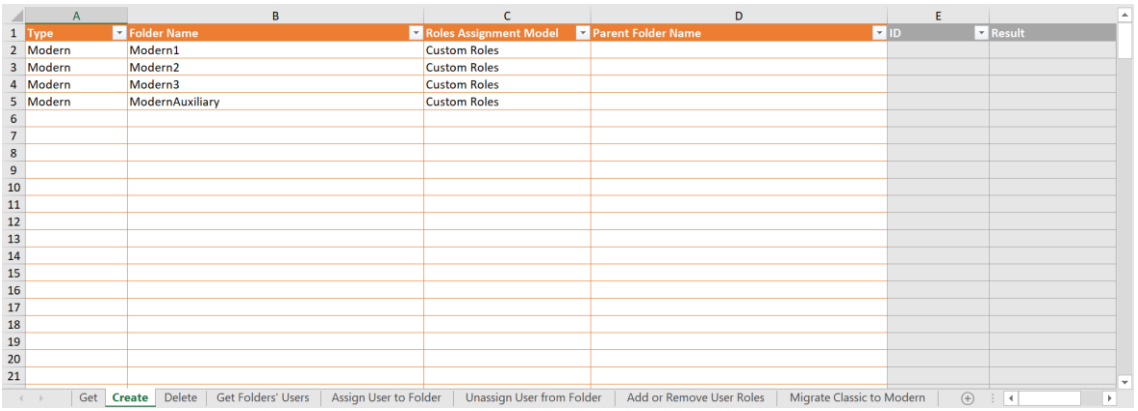



*Figure 19 - Configuration of Modern Folders After Migration*

After the conditions of User creation and Package update are fulfilled, the implementation of the described migration plan via Orchestrator Manager according the following steps:

1. Connect Orchestrator Manager and create the Modern Folders involved in the migration via the **Create** sheet of the **Folders.xlsx** workbook (Figure 20).

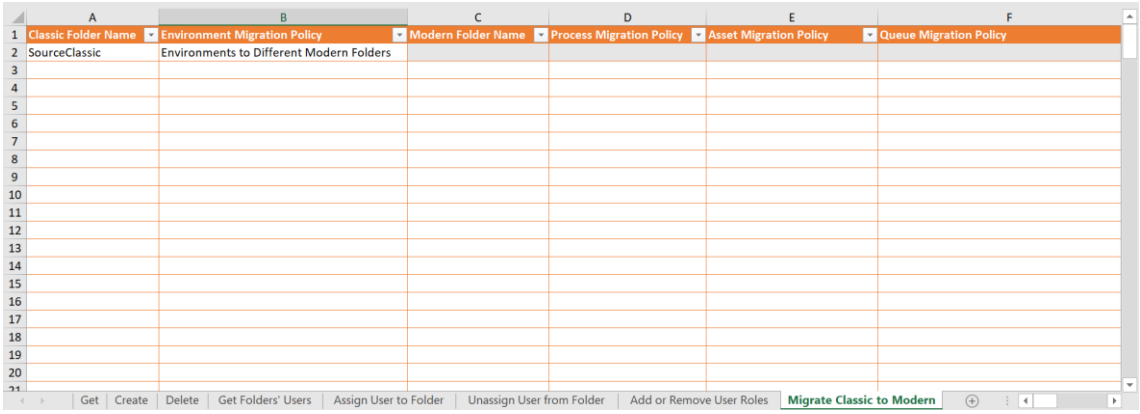


| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| 1 | Type | Folder Name | Roles Assignment Model | Parent Folder Name | ID | Result |
| 2 | Modern | Modern1 | Custom Roles | | | |
| 3 | Modern | Modern2 | Custom Roles | | | |
| 4 | Modern | Modern3 | Custom Roles | | | |
| 5 | Modern | ModernAuxiliary | Custom Roles | | | |

Get | Create | Delete | Get Folders' Users | Assign User to Folder | Unassign User from Folder | Add or Remove User Roles | Migrate Classic to Modern

*Figure 20 - Creation of Modern Folders*

2. In the **Migrate Classic to Modern** sheet of **Folders.xlsx**, input the name of the Classic Folder, *SourceClassic*, to be migrated and choose *Environments to Different Modern Folders* as Environment Migration Policy (Figure 21). It is not necessary to input the other fields. On Orchestrator Manager's Control Panel, choose the entity Folders and the operation Migrate Classic to Modern.

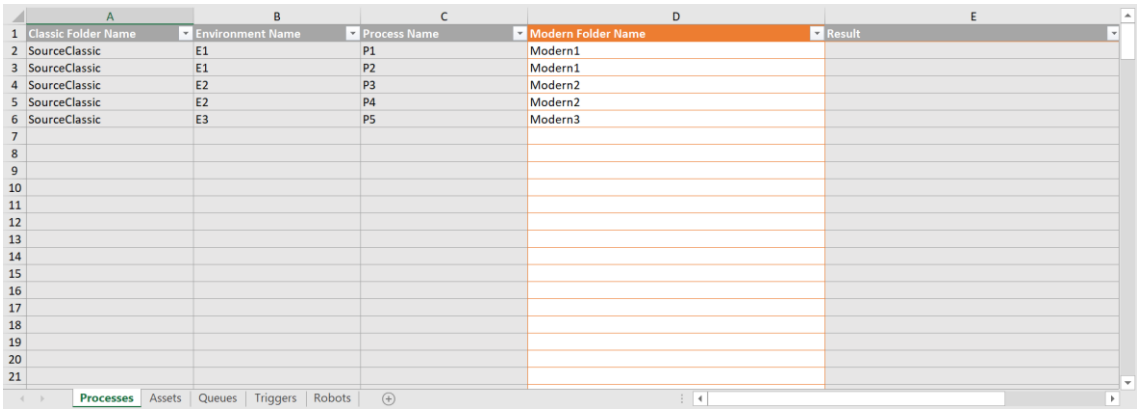


| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Classic Folder Name | Environment Migration Policy | Modern Folder Name | Process Migration Policy | Asset Migration Policy | Queue Migration Policy |
| 2 | SourceClassic | Environments to Different Modern Folders | | | | |

Get | Create | Delete | Get Folders' Users | Assign User to Folder | Unassign User from Folder | Add or Remove User Roles | Migrate Classic to Modern

*Figure 21 - Migrate Classic Modern Sheet*

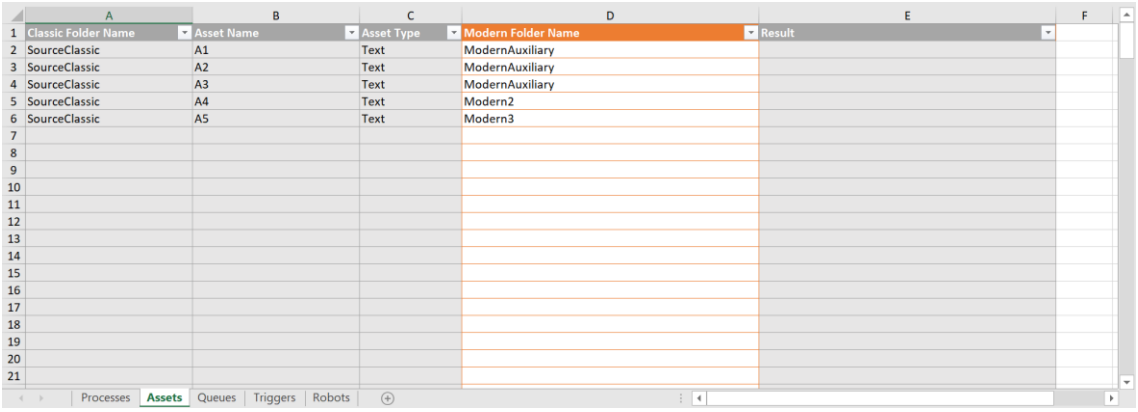
3. After *SourceClassic's* migration workbook is created and presented, input the names of the Modern Folders to which entities should be copied, as shown in Figure 22, Figure 23, Figure 24, Figure 25 and Figure 26. Note that, in Figure 26, rows for Robots *R1*, *R2* and *R3* were manually duplicated to account for the assignment of the corresponding Users to multiple Folders.
4. To execute the migration, confirm it using the Migration Panel (Figure 27). Orchestrator Manager will then copy entities to the specified Modern Folders and update the corresponding sheets with the results. Once it is done, the **Migrate Classic to Modern** sheet in **Folders.xlsx** is also updated.



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Classic Folder Name | Environment Name | Process Name | Modern Folder Name | Result |
| 2 | SourceClassic | E1 | P1 | Modern1 | |
| 3 | SourceClassic | E1 | P2 | Modern1 | |
| 4 | SourceClassic | E2 | P3 | Modern2 | |
| 5 | SourceClassic | E2 | P4 | Modern2 | |
| 6 | SourceClassic | E3 | P5 | Modern3 | |

Processes | Assets | Queues | Triggers | Robots

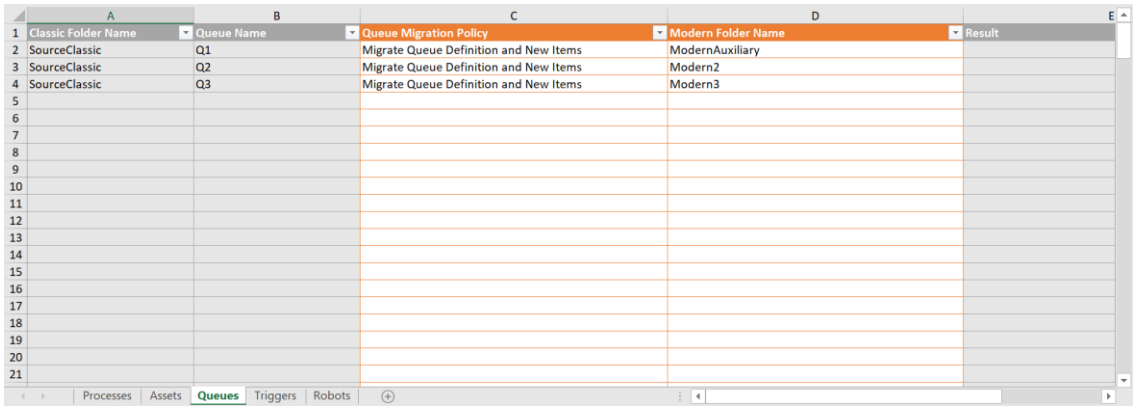
*Figure 22 - Migration of Processes to Modern Folders*



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Classic Folder Name | Asset Name | Asset Type | Modern Folder Name | Result | |
| 2 | SourceClassic | A1 | Text | ModernAuxiliary | | |
| 3 | SourceClassic | A2 | Text | ModernAuxiliary | | |
| 4 | SourceClassic | A3 | Text | ModernAuxiliary | | |
| 5 | SourceClassic | A4 | Text | Modern2 | | |
| 6 | SourceClassic | A5 | Text | Modern3 | | |

Processes | Assets | Queues | Triggers | Robots

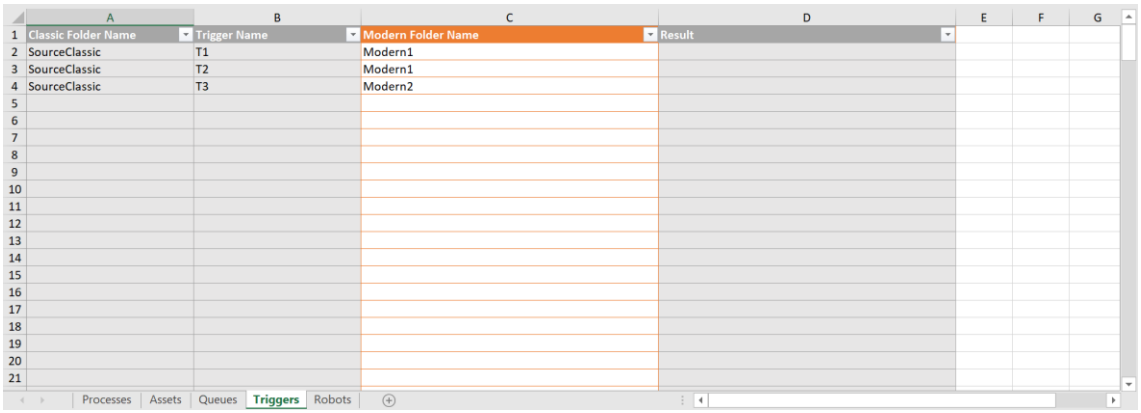
*Figure 23 - Migration of Assets to Modern Folders*



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Classic Folder Name | Queue Name | Queue Migration Policy | Modern Folder Name | Result |
| 2 | SourceClassic | Q1 | Migrate Queue Definition and New Items | ModernAuxiliary | |
| 3 | SourceClassic | Q2 | Migrate Queue Definition and New Items | Modern2 | |
| 4 | SourceClassic | Q3 | Migrate Queue Definition and New Items | Modern3 | |

Processes | Assets | Queues | Triggers | Robots

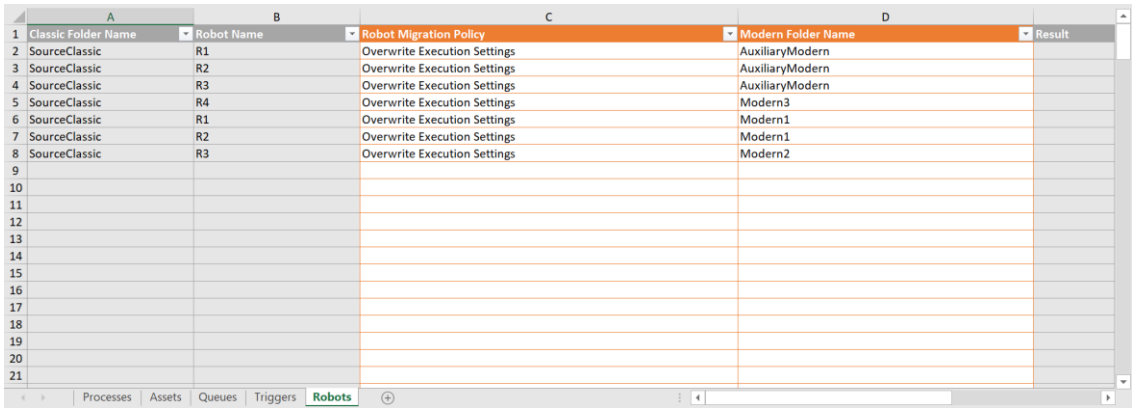
*Figure 24 - Migration of Queues to Modern Folders*



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Classic Folder Name | Trigger Name | Modern Folder Name | Result | | | |
| 2 | SourceClassic | T1 | Modern1 | | | | |
| 3 | SourceClassic | T2 | Modern1 | | | | |
| 4 | SourceClassic | T3 | Modern2 | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |
| 16 | | | | | | | |
| 17 | | | | | | | |
| 18 | | | | | | | |
| 19 | | | | | | | |
| 20 | | | | | | | |
| 21 | | | | | | | |

Processes | Assets | Queues | Triggers | Robots

*Figure 25 - Migration of Triggers to Modern Folders*



| | A | B | C | D | |
|---|---|---|---|---|---|
| 1 | Classic Folder Name | Robot Name | Robot Migration Policy | Modern Folder Name | Result |
| 2 | SourceClassic | R1 | Overwrite Execution Settings | AuxiliaryModern | |
| 3 | SourceClassic | R2 | Overwrite Execution Settings | AuxiliaryModern | |
| 4 | SourceClassic | R3 | Overwrite Execution Settings | AuxiliaryModern | |
| 5 | SourceClassic | R4 | Overwrite Execution Settings | Modern3 | |
| 6 | SourceClassic | R1 | Overwrite Execution Settings | Modern1 | |
| 7 | SourceClassic | R2 | Overwrite Execution Settings | Modern1 | |
| 8 | SourceClassic | R3 | Overwrite Execution Settings | Modern2 | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |
| 21 | | | | | |

Processes | Assets | Queues | Triggers | Robots

*Figure 26 - Migration of Robots to Modern Folders*

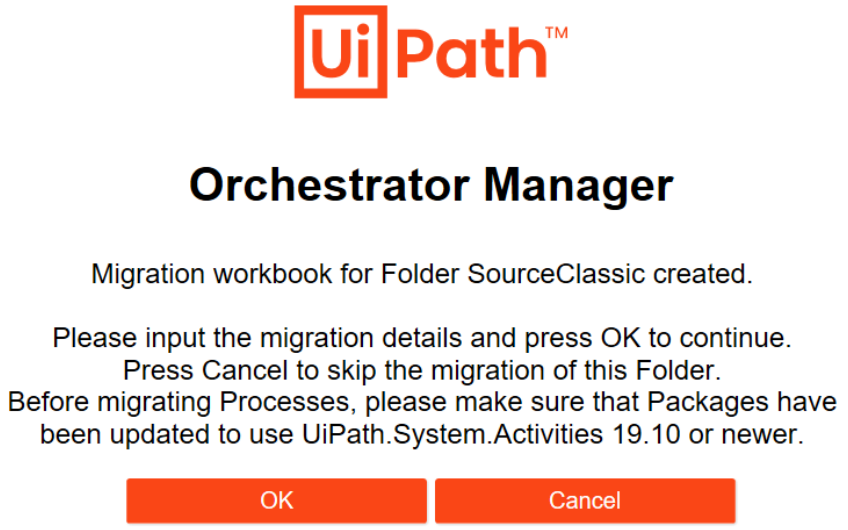



*Figure 27 - Migration Panel for SourceClassic*

The migration of Unattended Robots requires one more manual step, which is the assignment of Machine Templates to a target Modern Folder. For Processes to be executed by an Unattended Robot, there must be a Machine Template with available runtimes in the same Folder as the Process. A Machine Template can be assigned to a Folder via Orchestrator's web interface (Figure 28).

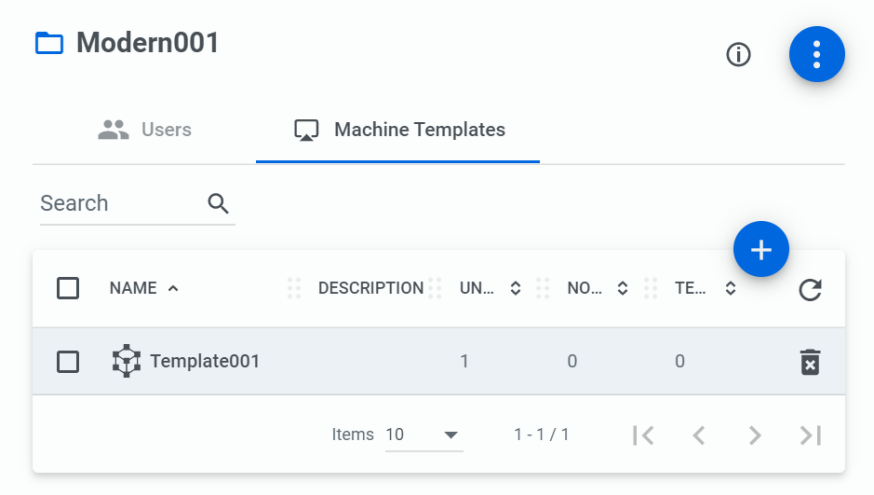



*Figure 28 - Assignment of Machine Templates to Folder*

Finally, note that some workflows might need to be updated after the migration to reflect changes in Folder path. In particular, activities such as *Get Asset*, *Get Credential*, *Add Queue Item* and *Get Transaction Item* need to specify the Modern Folder path via their *FolderPath* property[15]. When doing so, it is possible to specify absolute and relative paths for Folders, based on the Folder containing the Process that access entities in other Folders. For example, in the migration scenario described above, for Process *P1* to access Asset *A1*, the *FolderPath* property of the *Get Asset* activity used by Process *P1*'s Package uses *ModernAuxiliary* as value, indicating it refers to an Asset in the *ModernAuxiliary* Folder.

More details about migration to Modern Folders can be found at Orchestrator's online documentation[16].

[15] https://docs.uipath.com/activities/docs/get-queue-item#section-misc

[16] https://docs.uipath.com/orchestrator/docs/using-modern-folders#section-migrating-to-modern-folders

## Restrictions

The current version of Orchestrator Manager has the following limitations:

- The minimum Orchestrator version supported is 2018.4. It is required for Organization Units to be enabled if connecting to Orchestrator 2018.4 or Orchestrator 2019.4, even if there is only a single Organization Unit defined.
- Entities other than Assets, Environments, Machines, Processes, Robots, Users, Organization Units, Folders, Queues and Packages are currently not supported.
- The manipulation of Assets per Robot is not supported.

Moreover, Orchestrator Manager is mainly intended to be used for bulk operations, and it is by no means a replacement for Orchestrator's web interface.

## Distribution and Support

Orchestrator Manager is available as a UiPath Studio project and it can be downloaded from UiPath Connect[17].

This tool is offered under the UiPath Open Platform Activity License Agreement (available as the file **LICENSE.pdf** and also online[18]), and support is provided on a best-effort basis.

---

[17] https://connect.uipath.com/marketplace/components/orchestrator-manager

[18] https://www.uipath.com/hubfs/legalspot/UiPath_Activity_License_Agreement.pdf