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, Packages, Libraries and Triggers) with data defined in Microsoft 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).
- Migrating of entities from Classic Folders to Modern Folders.
- Migrating between tenants, including from on-premises tenants to tenants hosted on UiPath Automation Cloud.
- Listing Assets and Queues that are referenced by workflows of a given Package.
- Converting standard Machines to Machine templates.

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, Orchestrator 2020.10 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 operator 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 operator 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 is also used to input data in sheets that create, edit and delete entities.

In cases of operations other than **Get**, each row is processed 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 workbooks have different colors that represent cells that can be modified by the operator (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.

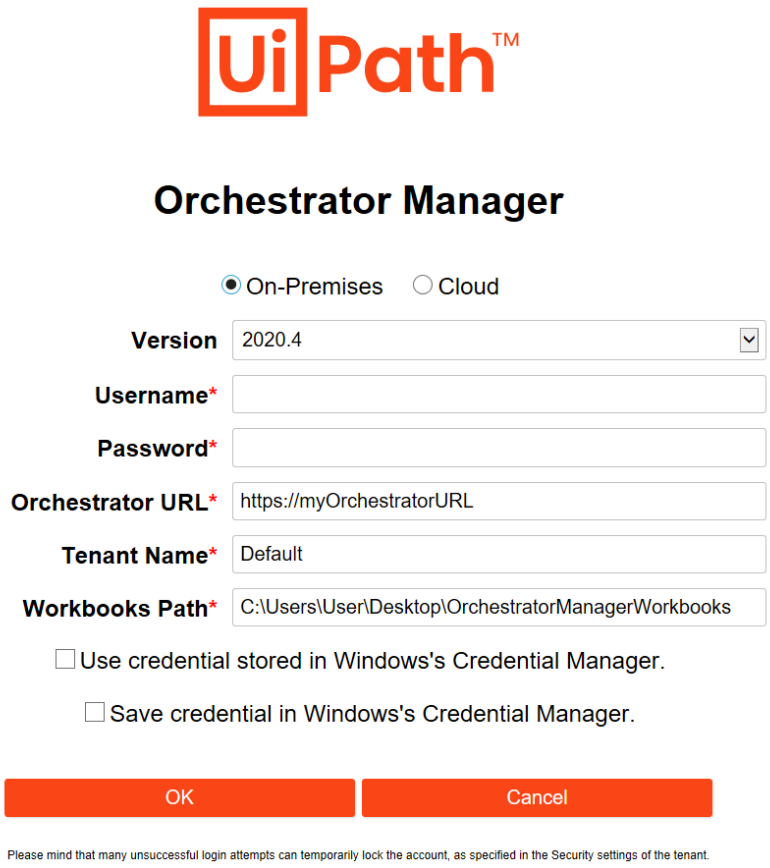


Figure 1 - Authentication Panel

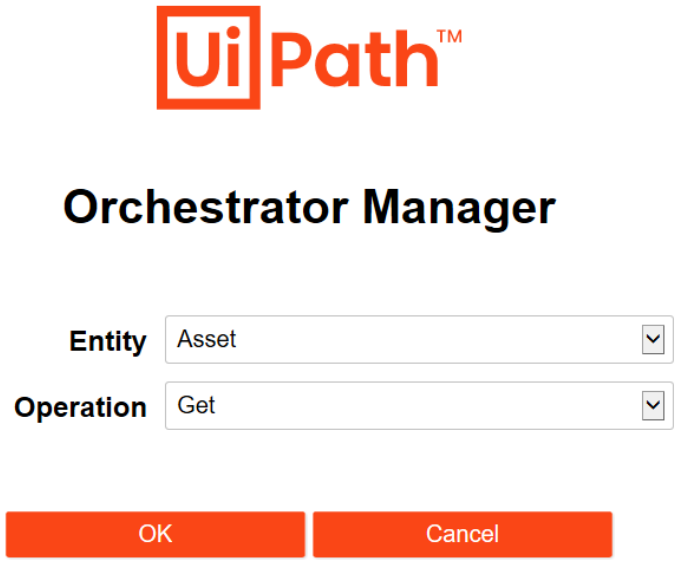


Figure 2 - Control Panel

In addition, different operations require different permissions to be set on Orchestrator. For instance, to create an Asset, it is necessary for the operator's Orchestrator User to have the *Assets.Create* permission. For this reason, it is recommended for operators 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.

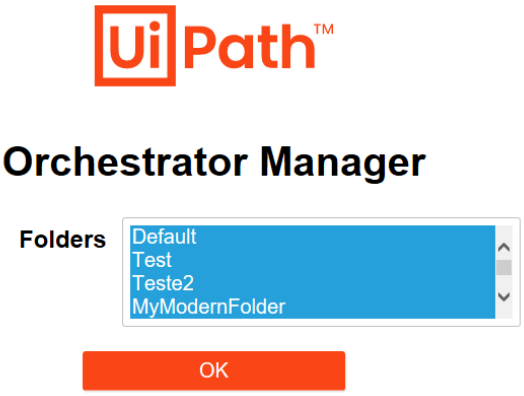


Figure 3 – Folders (Organization Units) Panel

Configuration

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

Settings

The first sheet, **Settings**, contains parameters used to connect to Orchestrator instances. They can be divided according to the deployment type of Orchestrator (i.e.,

[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

on-premises or UiPath Automation Cloud), except *EntitiesWorkbooksFolderPath*, which is 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 usually not necessary 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, *MaximumRequestsThreshold* and *RequestInterval*, 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.

FormsHeight and *FormsWidth* define the dimensions of the panels shown to the operator via the *Custom Input* activity.

MakeHTTPRequestNumberOfRetries, *MakeHTTPRequestRetryInterval* and *HTTPRequestActivityTimeout* are related to configurations of the HTTP Request activity and its use in the **Common\MakeHTTPRequest.xaml** workflow.

After that, there are many parameters that configure URLs and paths to local files and folders, such as *AutomationCloudURL* and *WorkbooksTemplatesPath*.

The parameter *PackagesMinimumDependenciesVersions* is used to check whether a Package uses the minimum version specified for a given dependency. This check is part of the procedure to confirm whether a Process is ready to be used in a Modern Folder.

AssetsReferringElements and *QueuesReferringElements* define XML elements that represent activities that relate to Assets and Queues. These names are used by **Get Referenced Entities** when inspecting Packages' workflows for references to Assets and Queues.

The parameters *AutomationCloudCredentialName* and *OnPremisesCredentialName* define the names of credentials to be optionally stored in Windows Credential Manager.

OrchestratorType is used to decide whether to show certain entities and allow certain operations that are exclusive to on-premises Orchestrator instances or UiPath Automation Cloud instances.

GetCredentialAssetsViaRobot 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 last few parameters are related to the migration of entities from Classic to Modern Folders, which is detailed in section Migration of Classic Folders to Modern Folders:

- *ShouldCreateBlankSpacePassword* is used if a credential Asset should be created, but a password is not specified. For example, when migrating Assets per Robot from a Classic Folder to a Modern Folder using the **Migrate Classic to Modern** operation on Folders, it is not possible to retrieve passwords even if *GetCredentialAssetsViaRobot* is set to *TRUE*. In such cases, *ShouldCreateBlankSpacePassword* can be used to allow the creation of Assets per User with a blank space as password value.
- *Enable202004FolderMigration* enables the migration of entities from Classic Folders to Modern Folders when using Orchestrator version 2020.4, which is not possible by default due to limitations of Orchestrator 2020.4 (i.e., lack of support for Assets per User and for deactivation of Robots).
- *EnableMigrationOfOldProcesses* allows the migration of Processes based on Packages that use versions lower than 19.10.1 of the UiPath.System.Activities dependency.
- *AddRolesAtTenantLevelDuringMigration* attempts to add a Robot's Roles to a User at tenant level.

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 executed by Orchestrator Manager, which are detailed in the following subsections.

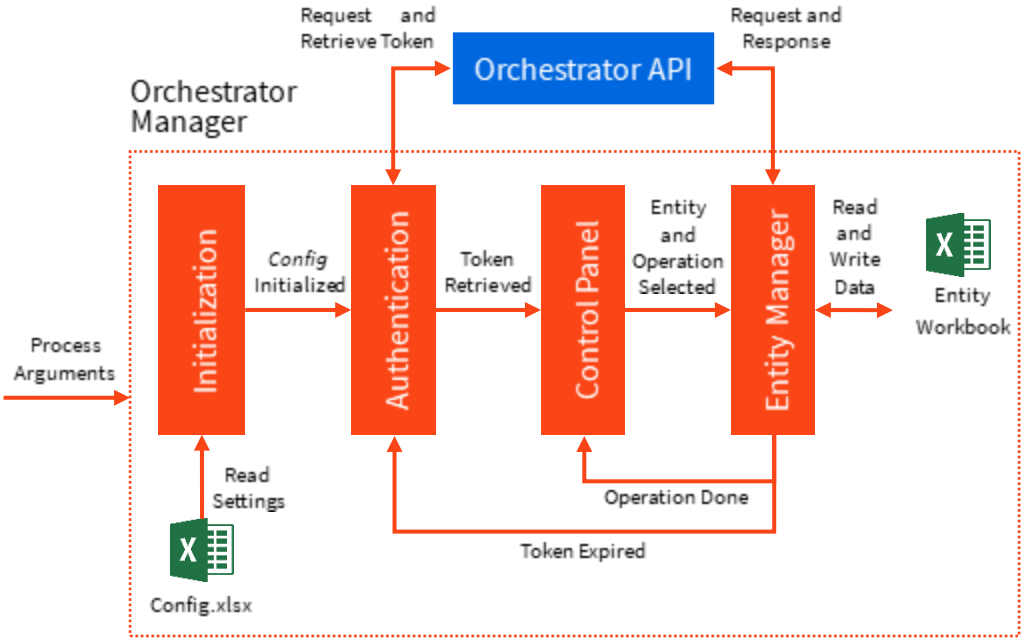


Figure 4 - Orchestrator Manager Architecture

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, receives data 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 to 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 workbook files. By default, the folder is created on the operator's Desktop folder.

The Authentication Panel is shown until a token is successfully retrieved or until the operator 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].

OAuth Authentication

For Cloud and On-Premises 2021.4 or later versions, Orchestrator Manager has the OAuth option available. To use OAuth Authentication, check "Use OAuth flow" option. To correctly authorize Orchestrator Manager, add application on the external applications page with **confidential application type** and add application scope with **OR.Administration** Orchestrator API access. After adding application, App ID and App Secret will be shown, which can be used for OAuth flow.

Control Panel

Once the authentication is successfully done, the operator 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

---

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

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

Orchestrator versions 2018.4 and 2019.4), Folders (for Orchestrator 2019.10 or newer), Queues, Packages, Libraries and Triggers (only for Orchestrator 2019.10 or newer). The available operations depend on the entity and are listed in section Entities' Workbooks.

Once the operator 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 the Control Panel for the operator to choose another operation. This cycle is repeated until the operator 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 operator 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's 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 operator, such as names, types and 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.

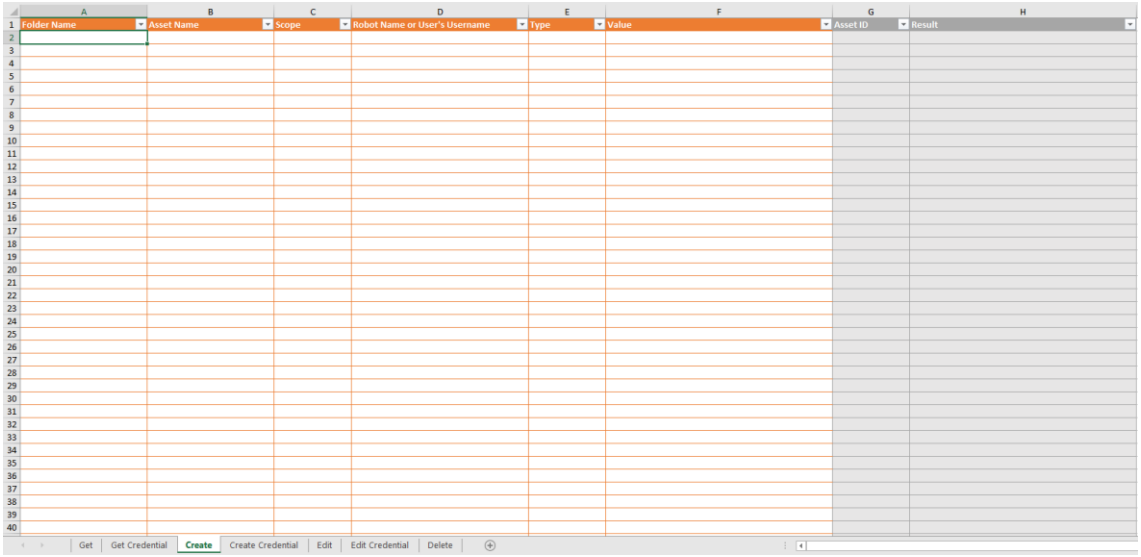


Figure 5 - Entity Workbook

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 - Libraries, Table 11 - Triggers and Table 12 - 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.

The option *Inherit from Tenant* is only supported by on-premises Orchestrator instances with version 2019.10 or 2020.4.

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. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. In the case of Assets per Robot or Assets per User, each Robot or User value is represented by an individual row.
Get Credential	Retrieves data about the existing global credential Assets. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. For security reasons, passwords of global 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: 1. The robot executing Orchestrator Manager needs to be connected to Orchestrator. 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. Since it is not possible to retrieve passwords of credential Assets per Robot and per User, the operation only retrieves global credential Assets.

Create	Creates Assets with the provided data. The field *Type* has a fixed number of possible values, according to Asset types available in Orchestrator: *Text*, *Bool* and *Integer*. The field *Scope* has a fixed number of possible values, according to Asset scopes available in Orchestrator: *Global*, *Robot* (only available in Classic Folders) and *User* (only available in Modern Folders). When creating Assets per Robot, use the field *Robot Name or User's Username* to specify the name of the Robot that should have access to the Asset's value being defined. The Robot must be in the same Folder as the Asset per Robot created. Values for multiple Robots must be specified in separate rows with the same Folder name, Asset name, scope and type. When creating Assets per User, use the field *Robot Name or User's Username* to specify the username of the User that should have access to the Asset's value being defined. The User must be assigned to the same Folder as the Asset per User created. Values for multiple Users must be specified in separate rows with the same Folder name, Asset name, scope and type. From Orchestrator version 2019.10, it is possible to define a default global value for an Asset per Robot (or per User). To do so, add another row using the same name as the Asset per Robot (or per User) being created and specify the scope *Global* to it.
Create Credential	Creates credential Assets with the provided data. The field *Scope* has a fixed number of possible values, according to Asset scopes available in Orchestrator: *Global*, *Robot* (only available in Classic Folders) and *User* (only available in Modern Folders).

	When creating credential Assets per Robot, it is necessary to specify the name of the Robot that should have access to the credential Asset's username and password being defined. The Robot must be in the same Folder as the Asset per Robot created. Values for multiple Robots must be specified in separate rows with the same Folder name, Asset name and scope. When creating Assets per User, it is necessary to specify the username of the User that should have access to the credential Asset's username and password being defined. The User must be assigned to the same Folder as the Asset per User created. Values for multiple Users must be specified in separate rows with the same Folder name, Asset name and scope. From Orchestrator version 2019.10, it is possible to define a default global value for an Asset per Robot (or per User). To do so, add another row using the same name as the Asset per Robot (or per User) being created and specify the scope *Global* to it.
Edit	Edits global Assets using the provided data. 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 empty are not modified. It is not possible to change the type of an Asset, and the new value must be compatible with the current type. Editing Assets per Robot and Assets per User is not supported.
Edit Credential	Edits credential Assets with the provided data. 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 empty are not modified.

	If *Username* is updated, a new password must also be provided. On the other hand, it is possible to update only *Password* and not input a new *Username*. Editing credential Assets per Robot and credential Assets per User is not supported.
Delete	Deletes the specified Assets. This operation supports both non-credential and credential Assets, as well as global Assets, Assets per Robot and Assets per User. When deleting an Asset per Robot or an Asset per User, it is enough to specify only one row, even if the Asset has multiple values. 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** or **Get Credential** operations on the Asset entity.
Link or Unlink	Link or unlink Assets to Folders. This operation is supported for cloud and on-premises 2021.4 or later versions.

Table 2 - Environments

Operation	Details
Get	Retrieves data about the existing Environments. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. Environments are only available in Classic Folders[8].
Create	Creates Environments with the provided data.

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

Delete	Deletes the specified Environments. 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. 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. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets.
Create	Creates Machines with the provided data. The field *Type* has a fixed number of possible values, according to Machine types available in Orchestrator: *Standard* and *Template*. Usually, a Machine key is automatically generated when a Machine is created, and the key is written to the **Create** sheet by Orchestrator Manager. However, in cases such as the conversion of standard Machines to Machine templates, it might be necessary to specify a Machine key that should be used when creating a Machine. This can be done by defining the key in the *Key* column, even if its color is gray.
Delete	Deletes the specified Machines. 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. This sheet is populated by Orchestrator Manager, and the operator 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. *Environment Name* should not be specified for Processes in Modern Folders. *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. 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.
Update to Specific Package	Update the specified Process to use the specific version of the Package.

Table 5 - Robots

Operation	Details
Get	Retrieves data about the existing Robots. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. 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. The field *Hosting Type* has a fixed number of possible values, according to hosting types available in Orchestrator: *Standard* and *Floating*. 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: • 2018.4: *NonProduction*, *Attended*, *Unattended* and *Development* • 2019.4: *NonProduction*, *Attended*, *Unattended* and *Development* • 2019.10: *NonProduction*, *Attended*, *Unattended*, *Development* and *StudioX* • 2020.4: *NonProduction*, *Attended*, *Unattended*, *Studio*, *Development*, *StudioX*, *Headless*, *StudioPro* and *TestAutomation* • 2020.10: *NonProduction*, *Attended*, *Unattended*, *Studio*, *Development*, *StudioX*, *Headless*, *StudioPro* and *TestAutomation*

	• UiPath Automation Cloud as of October 2020: *NonProduction*, *Attended*, *Unattended*, *Studio*, *Development*, *StudioX*, *Headless*, *StudioPro* and *TestAutomation* The possible values for each Orchestrator version can also be confirmed on Orchestrator's Swagger page[9]. Robots can only be created in Classic Folders. In the case of Modern Folders, Robots are associated with Users and do not represent independent entities.
Edit	Edits Robots using the provided data. 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 empty are not modified. It is not possible to change the *Hosting Type* of a Robot.
Delete	Deletes the specified Robots. 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. This sheet is populated by Orchestrator Manager, and the operator 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 from this operation, but they can be fetched via the **Get** operation on the Robot entity. 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. This operation is not available when connecting to UiPath Automation Cloud.
Create	Creates Users with the provided data. 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. The specification of Folders when creating Users is not supported by this operation, and it must be done via the **Assign User to Folder** operation on the Folder entity. The field *Password* must follow the security rules for passwords defined in the tenant's settings page on Orchestrator. Attended Robot, Unattended Robot, and Robot settings are only applicable to on-premises Orchestrator version 2019.10 or later. This operation is not available when connecting to UiPath Automation Cloud.
Edit	Edits Users using the provided data. 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 empty are not modified. It is not possible to change the *Username* of the User. The field *Status* has a fixed number of possible values, according to hosting types available in Orchestrator: *Active* and *Inactive*. Attended Robot, Unattended Robot, and Robot settings are only

	applicable to on-premises Orchestrator version 2019.10 or later. 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**. For changes regarding User assignment to Folders and User Roles at Folder level, refer to operations on the Folder entity. This operation is not available when connecting to UiPath Automation Cloud.
Delete	Deletes the specified Users. 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. 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. The ID of a User can be retrieved by the **Get** operation on the User entity. 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. For changes regarding User Roles at Folder level, refer to the operation **Add or Remove User Roles** on the Folder entity. 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. 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. For changes regarding User assignment to Folder, refer to the operations **Assign User to Folder and Unassign User from Folder** on the Folder entity. 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. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. 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. 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. 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. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. Only data about Queues' definitions is retrieved. Use the **Download Queue Items** operation to retrieve Queue items.
Create	Creates Queues with the provided data. 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. 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. A Queue can be deleted even if it contains Queue items.
Download Queue Items	Downloads Queue items from the specified Queue. The field *Download Folder Path* indicates the full path of a local folder to which Queue items should be saved. Items of each Queue are downloaded as a CSV file that follows the same format as the one 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. 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 format of the columns of the CSV file must follow the format accepted by Orchestrator[10] or the exported format as below. The CSV file downloaded via the **Download Queue Items** operation has a particular format which will be automatically converted to the format that can be used by the **Upload Queue Items**. The field *Commit Type* has a fixed number of possible values, according to types available in Orchestrator[11]: *AllOrNothing*, *StopOnFirstFailure* and *ProcessAllIndependently*. This operation is not available in the case of on-premises Orchestrator instances with version 2018.4.
Link or Unlink	Link or unlink Queues to Folders. This operation is supported for cloud and on-premises 2021.4 or later versions.

[10] https://docs.uipath.com/orchestrator/docs/about-queues-and-transactions#column-mapping

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

Table 9 - Packages

Operation	Details
Get	Retrieves data about the existing Packages. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. Each Package version is displayed in a separate row of the **Get** sheet.
Delete	Deletes the specified Packages. To prevent accidental deletion, it is necessary to provide *Name*, *Version* and *Key* of each Package to be deleted. This data can be retrieved by the **Get** operation on the Package entity.
Download	Downloads the specified Packages. 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. Full paths of successfully downloaded Packages are written to the field *Download Package File Path*, and they can be reused by the **Upload** operation. Package files are overwritten if they already exist in the specified folder.
Upload	Uploads the specified Packages. 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.

	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. 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: • Workflow Name: Name of the workflow that references an Asset or a Queue. • Hierarchy to Activity: Names of activities in the workflow that are ancestors of the activity that references an Asset or a Queue. • 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). • 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. • 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 [12] and uses the **Config.xlsx** file to store references to Assets and Queues, this file is also checked for the actual name of entities.

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

Table 10 - Libraries

Operation	Details
Get	Retrieves data about the existing Libraries. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. Each Library version is displayed in a separate row of the **Get** sheet.
Delete	Deletes the specified Triggers. To prevent accidental deletion, it is necessary to provide *Name*, *Version* and *Key* of each Library to be deleted. This data can be retrieved by the **Get** operation on the Package entity.
Download	Downloads the specified Libraries. The field *Download Folder Path* indicates the full path to a folder to which Libraries should be saved. Libraries are downloaded as .nupkg files, following the same format as the ones downloaded via Orchestrator's web interface. Full paths of successfully downloaded Libraries are written to the field *Download Library File Path*, and they can be reused by the **Upload** operation. Libraries files are overwritten if they already exist in the specified folder.
Upload	Uploads the specified Libraries. The field *Library File Path* indicates the full path to a file of type .nupkg. The name and the version of the Library are automatically defined based on the metadata included in the Library file.

Table 11 - Triggers

Operation	Details
Get	Retrieves data about the existing Triggers. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer.
Create	Creates Triggers with the provided data. Due to the large number of fields necessary to create a Trigger, instead of manually entering the data, it is expected that the **Create** operation is mostly used based on data retrieved by the **Get** operation, such as when copying Triggers from one Folder to another. The Process started by the trigger is specified by the field *Process Name*, and it is necessary for the Process to be available in the same Folder as the Trigger being created. In the case of Queue Triggers, it is also necessary for the Queue, specified by the field *Queue Name*, to be in the same Folder as the Trigger. *Queue Triggers* also require the definition of the fields *Items Activation Threshold*, *Items Per Job Activation* and *Max Jobs for Activation*. The fields *Enabled* and *Use Calendar* can have two values: *TRUE* or *FALSE*. *Input Arguments Definition* can be defined by a JSON string with key-value pairs, where keys represent names of arguments and values represent their values. The field *Start Strategy* is specified according to the following possible values:

	• *-1* indicates that all Robots should be used. This option is not supported by Modern Folders. • *0* indicates that specific Robots should be used. This option is not currently supported by Orchestrator Manager. • Values greater than *0* indicate the number of times the Process should be executed using dynamic allocation. This is the only supported option in the case of Modern Folders. The field *Stop Strategy* can be specified as either *SoftStop* or *Kill*, depending on how the Job created by the Trigger should be stopped. *Job Priority* accepts three values, which indicate the priority of the Job created by the Trigger: *Low*, *Normal* or *High*. If the field *Calendar Name* is used, the specified Calendar must exist in the tenant before the Trigger is created. The possible values for other fields can be confirmed on Orchestrator's Swagger page. 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 Triggers. To prevent accidental deletion, it is necessary to provide both ID and name of the Folder containing the Trigger, as well as ID and name of each Trigger to be deleted. This data can be retrieved by the **Get** operation on the Trigger entity. This operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer.
Enable or Disable	Enable or disable the specified Triggers. This operation is only available when connecting to UiPath

Triggers	Automation Cloud or on-premises Orchestrator instances with version 2019.10 or newer.

Table 12 - Folders

Operation	Details
Get	Retrieves data about the existing Folders. This sheet is populated by Orchestrator Manager, and the operator is not required to input any data. The retrieved data can be copied for use in other sheets. Note that Classic Folders or first level Modern Folders do not have parent Folders. 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. If the Folder to be created 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. It is not possible to specify parent Folders for Classic Folders. The field *Type* has a fixed number of possible values, according to Folder types available in Orchestrator: *Classic* and *Modern*. 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.

	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. 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. 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 operator is not required to input any data. The retrieved data can be copied for use in other sheets. 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. 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. 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 whose Role Assignment Model is *Inherit from Tenant*. The field *Roles Names* is ignored in the case of Classic Folders, as it is not possible to assign Roles to Users at Folder level.

	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. If this operation is executed multiple times, the User is assigned with only the Roles of the latest execution. In other words, the assignment of Roles is not cumulative. 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. 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* 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. This operation is only supported by Modern Folders. 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.

	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*. For changes regarding User Roles at tenant level, refer to the operation **Add or Remove Roles** 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.
Assign or Unassign Machines	Assigns Machines to Folders or unassigns Machines from Folders. This operation is only supported by Modern Folders. When assigning, set *Assign* to *TRUE*. When unassigning, set *Assign* to *FALSE*. 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 entities of the specified Classic Folder, according to the selected migration policies. The field *Environment Migration Policy* has a fixed number of values: *Environments to Different Modern Folders* and *Environments to Same Modern Folder*. 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 in **Folders.xlsx**, as a new migration workbook for the specified Folder is created and shown to the operator. Details of migration workbooks can be found in Table 13. If *Environments to Same Modern Folder* is selected, all

	Environments and related entities 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 entities:
	• Process Migration Policy - Determines whether Processes in the Classic Folder should be created in the Modern Folder, according to the following possible values:
	  ○ *Migrate All Processes*: Copies all Processes to the target Modern Folder. Modern Folders require that Packages used to create Processes reference UiPath.System.Activities version 19.10.1 or newer, and a warning log message is generated in the case of Processes using Packages with older versions of that dependency.
	  ○ *Do Not Migrate Processes*: No Processes are copied to the target Modern Folder.
	• Asset Migration Policy - Determines how Assets should be created in the Modern Folder, according to the following possible values:
	  ○ *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. Assets per Robot are created as Assets per User in the Modern Folder, which requires that the corresponding Users are created beforehand and have Robots assigned to them, as shown in Figure 14.
	  ○ *Migrate Non-Credential Assets*: Only copies Assets of type Text, Integer and Bool. Assets per Robot

	are created as Assets per User in the Modern Folder, which requires that the corresponding Users are created beforehand and have Robots assigned to them, as shown in Figure 14. ○ *Do Not Migrate Assets*: No Assets are copied to the target Modern Folder. • Queue Migration Policy - Determines how Queues and Queue items should be created in the Modern Folder, according to the following possible values: ○ *Migrate Queue Definitions and New Items*: Copies all data about Queues, including Queue items with status *New* (the 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. ○ *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. ○ *Do Not Migrate Queues*: No Queues are copied to the target Modern Folder. • Trigger Migration Policy - Determines whether Triggers should be created in the Modern Folder, according to the following possible values: ○ *Migrate All Triggers*: Copies all Triggers to the target Modern Folder and disables them in the source Classic 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. ○ *Do Not Migrate Triggers*: No Triggers are copied to the target Modern Folder. • 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, as shown in Figure 14. Furthermore, the Robot provisioned 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: ○ *Overwrite Execution Settings*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder and replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder. The Robot in the Classic Folder remains active. ○ *Deactivate Classic Robots*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder and deactivates the Robot in the Classic Folder. Execution settings are not overwritten. ○ *Overwrite Execution Settings and Deactivate Classic Robots*: Assigns the User corresponding to

	the Robot to be migrated to the target Modern Folder, replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder, and deactivates the Robot in the Classic Folder. ○ *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. ○ *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. Other than the above, there are two additional details regarding the migration of Robots: • When a User corresponding to a Robot to be migrated is assigned to a Modern Folder, the User receives, at Folder level, all Roles that the Robot in the Classic Folder had. The parameter *AddRolesAtTenantLevelDuringMigration* (**Advanced Settings** sheet of **Config.xlsx**) can be used if it is necessary to assign these Roles at tenant level as well. Finer adjustments related to Roles can be done via Orchestrator's web interface or using the operations **Add or Remove User Roles** on the Folder entity and **Add or Remove Roles** on the Users entity. • 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. Error or warnings that occur during 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* column of **Folders.xlsx**. By default, the **Migrate Classic to Modern** operation is only available when connecting to UiPath Automation Cloud or on-premises Orchestrator instances with version 2020.10 or newer. If it is necessary to migrate Folders in the case of Orchestrator version 2020.4, the parameter *Enable202004FolderMigration* to in the **Advanced Settings** sheet of **Config.xlsx** must be set to *TRUE*. Note that Orchestrator 2020.4 does not support deactivation of Robots in Classic Folders nor the creation of Assets per User.

Table 13 – Migration from Classic to Modern

Entity	Details
Processes	Specifies what Processes should be copied to what Modern Folders. Modern Folders require that Packages used to create Processes reference UiPath.System.Activities version 19.10.1 or newer, and a warning log message is generated in the case of Processes using Packages with older versions of that dependency. 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. 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. 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. Assets per Robot are created as Assets per User in the Modern Folder, which requires that the corresponding Users are created beforehand and have Robots assigned to them, as shown in Figure 14.
Queues	Specifies what Queues should be copied to what Modern Folders. It is possible to specify the migration policy, according to the following values: • *Migrate Queue Definition and New Items*: Copies all data about the Queue, including Queue items with status *New* (the 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. • *Migrate Queue Definition But Not Queue Items*: Only copies data about the Queue definition and does not copy Queue items. 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. 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. After migrated, the Trigger in the Classic Folder is disabled. Since Queue Triggers have Processes associated with them, they must be copied after the corresponding Process is available in the target Modern Folder. Migrated Time Triggers utilize Dynamic Allocation as their Execution Target option.

	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 should be migrated to what Modern Folders. 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, as shown in Figure 14. Moreover, 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: • *Overwrite Execution Settings*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder and replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder. The Robot in the Classic Folder remains active. • *Deactivate Classic Robots*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder and deactivates the Robot in the Classic Folder. Execution settings are not overwritten. • *Overwrite Execution Settings and Deactivate Classic Robot*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder, replaces execution settings defined for the Robot provisioned at User level with the settings defined for the Robot in the Classic Folder, and deactivates the Robot in the Classic Folder.

	• *Do Not Overwrite Execution Settings Nor Deactivate Classic Robot*: Assigns the User corresponding to the Robot to be migrated to the target Modern Folder, but the execution settings of the User's associated Robot are not replaced, and the Robot in the Classic Folder is not deactivated. When a User corresponding to a Robot to be migrated is assigned to a Modern Folder, the User receives, at Folder level, all Roles that the Robot in the Classic Folder had. The parameter *AddRolesAtTenantLevelDuringMigration* (**Advanced Settings** sheet of **Config.xlsx**) can be used if it is necessary to assign these Roles at tenant level as well. Finer adjustments related to Roles can be done via Orchestrator's web interface or using the operations **Add or Remove User Roles** on the Folder entity and **Add or Remove Roles** on the Users entity. 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. If a User corresponding to a Robot needs to be migrated to multiple Modern Folders, it is necessary to duplicate the row corresponding to that Robot and specify different target Modern Folders.

Examples of Use

This section present 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 create 30 Floating Attended Robots in a Classic Folder, equally distributed among three Environments: Finance, HR and Legal.

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 use case highlighted 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 administrators 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[13].

---

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

	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	E	F	G	H	I
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									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									

Get | Create | Delete | Add or Remove Robots

Figure 7 - Create Environments

	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

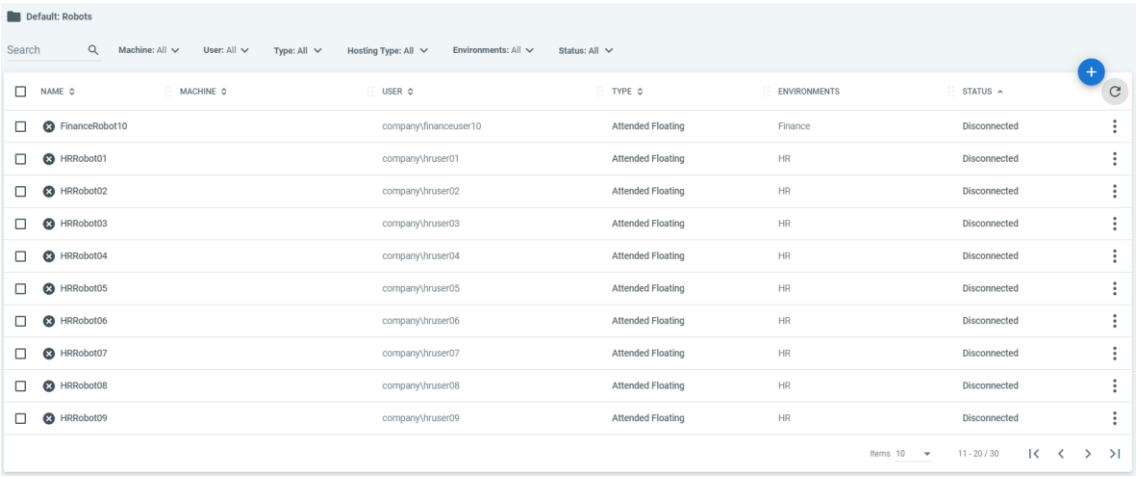


Figure 9 – Robots Provisioned and Added to Environments

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[14], or when copying entities from a tenant used for development purposes to a tenant used

[14] https://docs.uipath.com/automation-cloud/docs/migrating-data-from-on-premises-to-cloud-orchestrator

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.

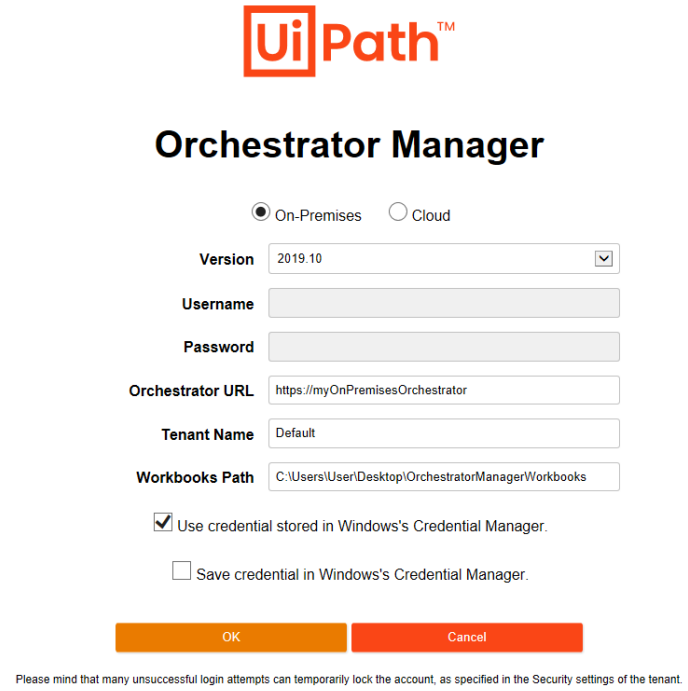
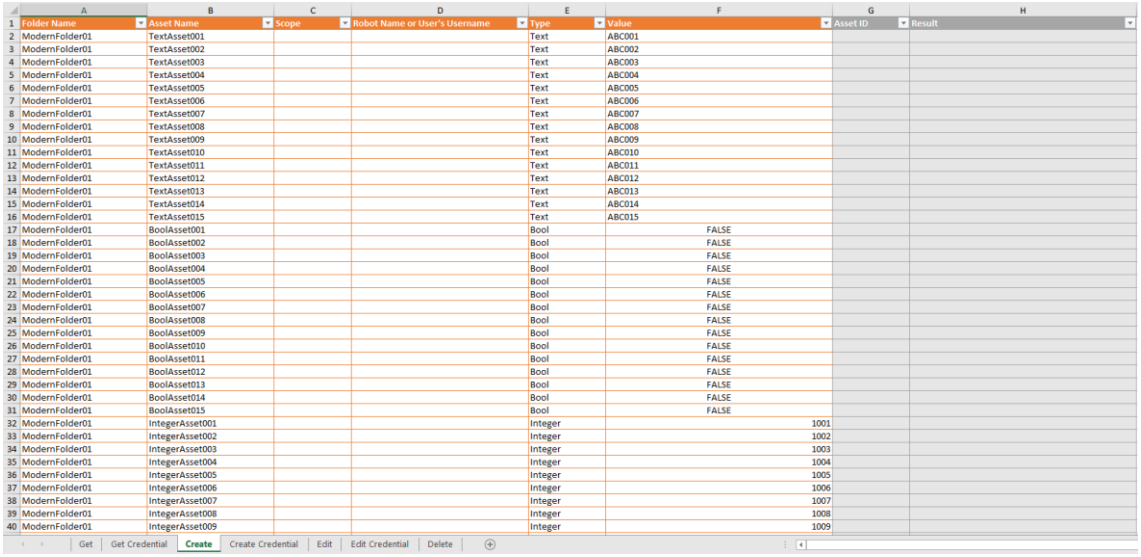


Figure 10 - Connection to On-Premises Source Tenant

	D	E	F	G	H	I
1	Asset Name	Scope	Robot ID or User ID	Robot Name or User's Username	Type	Value
2	TextAsset001	Global			Text	ABC001
3	TextAsset002	Global			Text	ABC002
4	TextAsset003	Global			Text	ABC003
5	TextAsset004	Global			Text	ABC004
6	TextAsset005	Global			Text	ABC005
7	TextAsset006	Global			Text	ABC006
8	TextAsset007	Global			Text	ABC007
9	TextAsset008	Global			Text	ABC008
10	TextAsset009	Global			Text	ABC009
11	TextAsset010	Global			Text	ABC010
12	TextAsset011	Global			Text	ABC011
13	TextAsset012	Global			Text	ABC012
14	TextAsset013	Global			Text	ABC013
15	TextAsset014	Global			Text	ABC014
16	TextAsset015	Global			Text	ABC015
17	BoolAsset001	Global			Bool	FALSE
18	BoolAsset002	Global			Bool	FALSE
19	BoolAsset003	Global			Bool	FALSE
20	BoolAsset004	Global			Bool	FALSE
21	BoolAsset005	Global			Bool	FALSE
22	BoolAsset006	Global			Bool	FALSE
23	BoolAsset007	Global			Bool	FALSE
24	BoolAsset008	Global			Bool	FALSE
25	BoolAsset009	Global			Bool	FALSE
26	BoolAsset010	Global			Bool	FALSE
27	BoolAsset011	Global			Bool	FALSE
28	BoolAsset012	Global			Bool	FALSE
29	BoolAsset013	Global			Bool	FALSE
30	BoolAsset014	Global			Bool	FALSE
31	BoolAsset015	Global			Bool	FALSE
32	IntegerAsset001	Global			Integer	1001
33	IntegerAsset002	Global			Integer	1002
34	IntegerAsset003	Global			Integer	1003
35	IntegerAsset004	Global			Integer	1004
36	IntegerAsset005	Global			Integer	1005
37	IntegerAsset006	Global			Integer	1006
38	IntegerAsset007	Global			Integer	1007
39	IntegerAsset008	Global			Integer	1008
40	IntegerAsset009	Global			Integer	1009

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

Figure 11 - Get Assets from Source Tenant



	A	B	C	D	E	F	G	H
1	Folder Name	Asset Name	Scope	Robot Name or User's Username	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	BoolAsset001			Bool	FALSE		
18	ModernFolder01	BoolAsset002			Bool	FALSE		
19	ModernFolder01	BoolAsset003			Bool	FALSE		
20	ModernFolder01	BoolAsset004			Bool	FALSE		
21	ModernFolder01	BoolAsset005			Bool	FALSE		
22	ModernFolder01	BoolAsset006			Bool	FALSE		
23	ModernFolder01	BoolAsset007			Bool	FALSE		
24	ModernFolder01	BoolAsset008			Bool	FALSE		
25	ModernFolder01	BoolAsset009			Bool	FALSE		
26	ModernFolder01	BoolAsset010			Bool	FALSE		
27	ModernFolder01	BoolAsset011			Bool	FALSE		
28	ModernFolder01	BoolAsset012			Bool	FALSE		
29	ModernFolder01	BoolAsset013			Bool	FALSE		
30	ModernFolder01	BoolAsset014			Bool	FALSE		
31	ModernFolder01	BoolAsset015			Bool	FALSE		
32	ModernFolder01	IntegerAsset001			Integer	1001		
33	ModernFolder01	IntegerAsset002			Integer	1002		
34	ModernFolder01	IntegerAsset003			Integer	1003		
35	ModernFolder01	IntegerAsset004			Integer	1004		
36	ModernFolder01	IntegerAsset005			Integer	1005		
37	ModernFolder01	IntegerAsset006			Integer	1006		
38	ModernFolder01	IntegerAsset007			Integer	1007		
39	ModernFolder01	IntegerAsset008			Integer	1008		
40	ModernFolder01	IntegerAsset009			Integer	1009		

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

Figure 12 - Details of Assets to Create

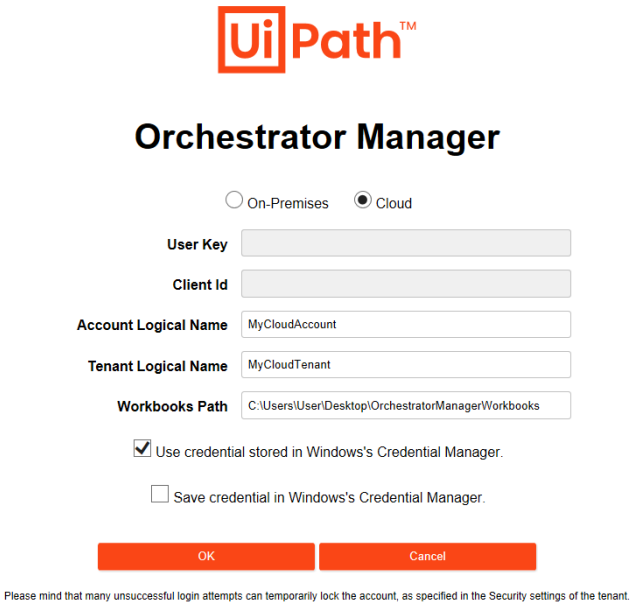


Figure 13 - Connection to Cloud Target Tenant

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, as with the migration between tenants, manually copying all entities from a Classic Folder to a Modern Folder can also be a very time-consuming task.

Orchestrator Manager includes the operation Migrate Classic to Modern on the Folder

entity (Table 12), which can be used to accelerate the migration procedure and reduce the risk of mistakes.

Before starting the migration, the following should be considered:

1. To fully take advantage of Modern Folders' features, Processes should use Packages that have version 19.10.1 or newer of the dependency UiPath.System.Activities. For this reason, it is recommended to republish Packages using the latest versions of their dependencies and update their corresponding Processes before starting the migration. By default, Orchestrator Manager copies Processes even if they use outdated Packages, but it is possible to prevent the migration of older Processes by setting the parameter *EnableMigrationOfOldProcesses* (**Advanced Settings** sheet of **Config.xlsx**) to *FALSE*.
2. Modern Folders do not defined Robots as an independent entity. Instead, they are associated with Users at tenant level[15], and those Users should be included in the appropriate Folders[16]. Because of this, it is necessary to setup Users before the migration, adding new Users if necessary and enabling their Robots with the appropriate usernames and Robot settings. 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 (Figure 14) must match the username used by the Robot migrated from the Classic Folder (Figure 15).

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

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

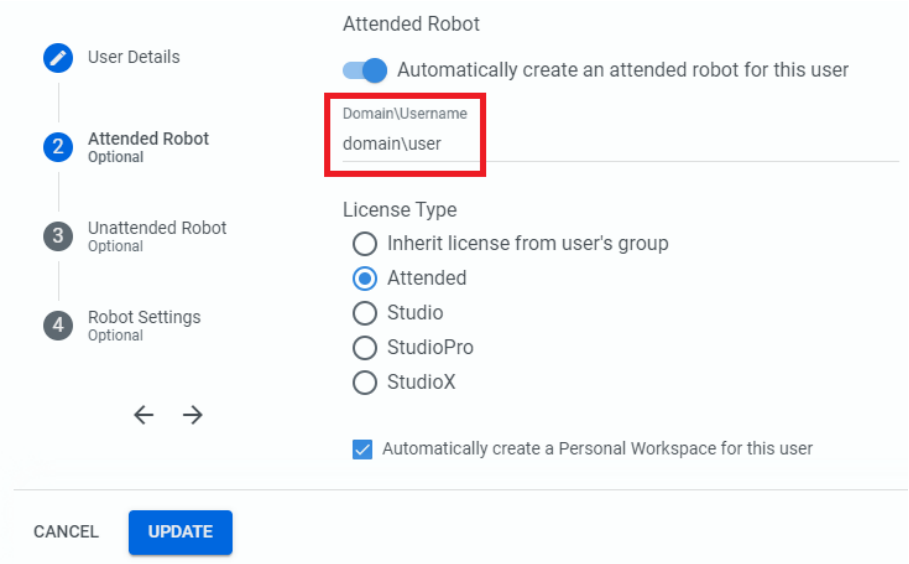


Figure 14 - Enabling a User's Attended Robot

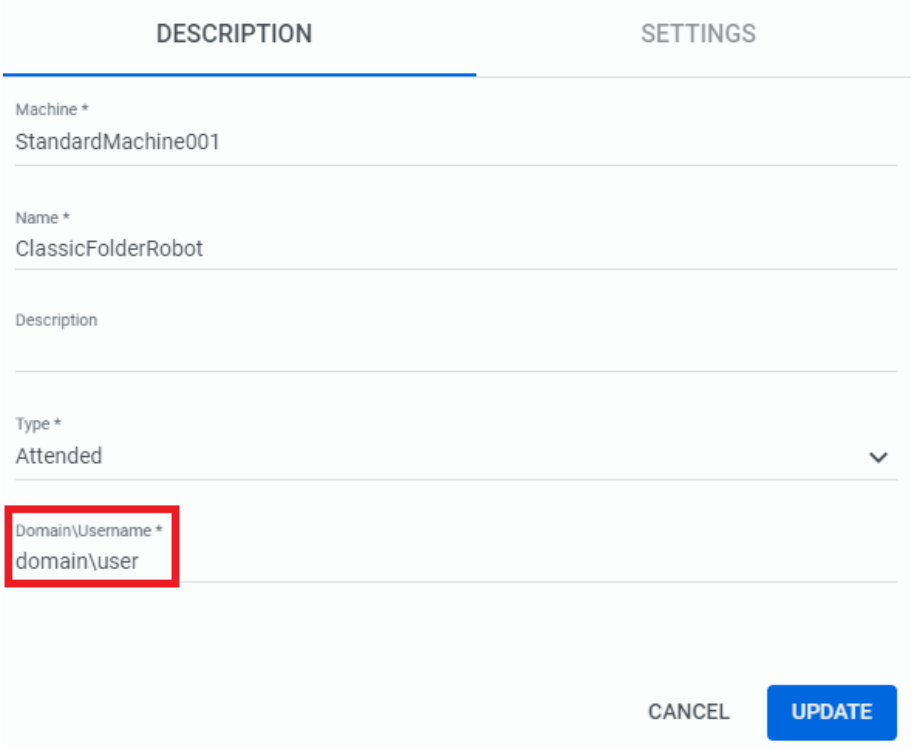


Figure 15 - Details of Robot in Classic Folder

Before copying entities from Classic to Modern Folders, an important step is to create a migration plan that gives a clear view of how entities should be divided among multiple Modern Folders.

In particular, this plan should specify how multiple Classic Folders should be migrated,

and also, for each Classic Folder, what should be done with Environments, their Processes, and entities (i.e., Assets and Queues) referenced by such Processes.

One common approach is to have one Classic Folder mapped into one root (i.e., first-level) Modern Folder. In such cases, there are two common options regarding how Environments should be handled:

1. All Environments in the Classic Folder are mapped into the root Modern Folder. This can be a good alternative if there is only a single Environment defined in the Classic Folder or if all Users (and their associated Robots) assigned to the target Modern Folder require access to all entities migrated.
2. Each Environment in the Classic Folder is mapped to a different subfolder of the root 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 divided among subfolders, other than one subfolder per Environment, there can also be additional subfolders to store shared entities. These entities can be accessed by Processes in different subfolders via the *FolderPath* property of activities such as *Get Asset* or *Get Transaction Item*.

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 12. For the second option, Orchestrator Manager makes use of a dedicated migration workbook for each Classic Folder to be migrated (Table 13).

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 16. *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*).

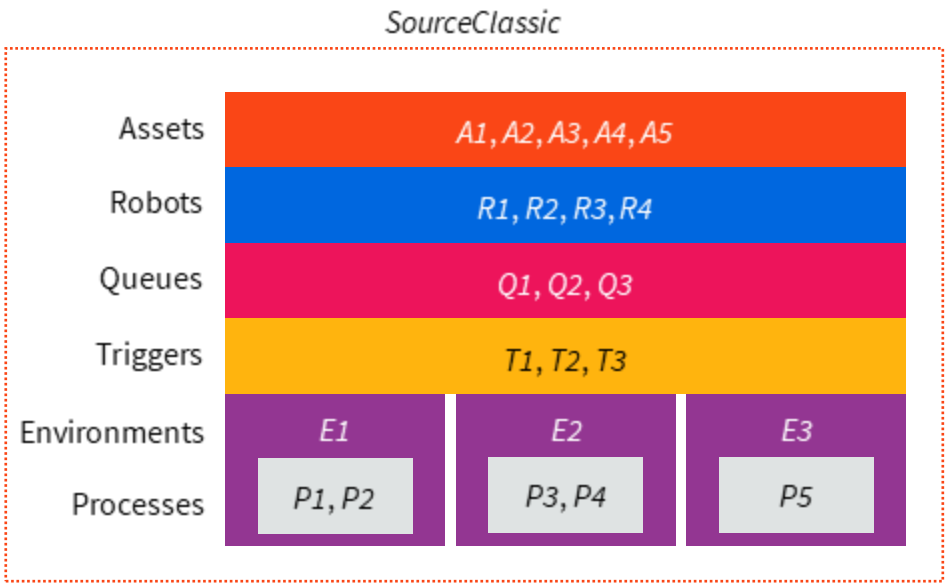


Figure 16 - Example of Classic Folder SourceClassic

The relationships between these entities are represented in the following figures. For instance, Figure 17 shows that Process *P1* utilizes Assets *A1* and *A3* as well as Queue *Q1*; Figure 18 depicts Robots *R1* and *R2* assigned to Environment *E1*; and Figure 19 illustrates that Time Trigger *T1* triggers Process *P1*.

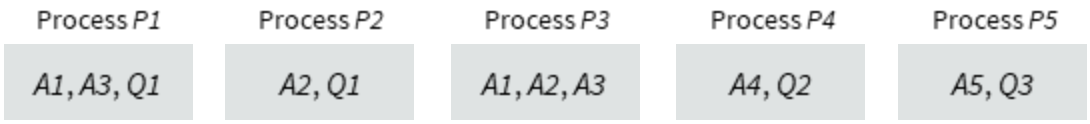


Figure 17 – Utilization of Assets and Queues by Processes in Classic Folder SourceClassic

Figure 18 – Assignment of Robots to Environments in Classic Folder SourceClassic

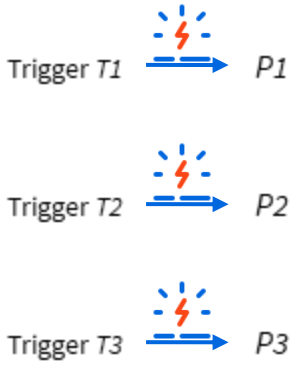


Figure 19 – Relationship Between Triggers and Triggered Processes in Classic Folder SourceClassic

Based on such relationships and the decision to have Environments mapped into different subfolders, a possible configuration using Modern Folders can be seen in Figure 20 and it is detailed as follows:

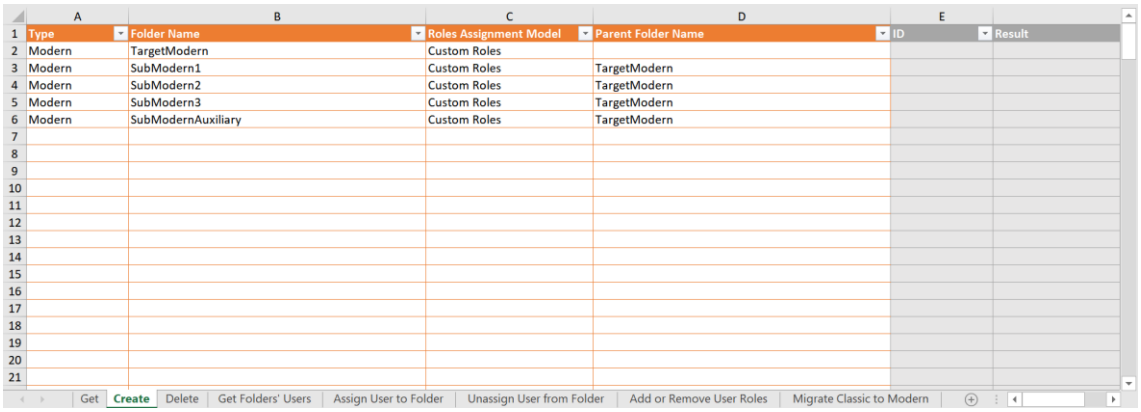
- It is assumed that Users were created beforehand and enabled with Robots for each Robot in the Classic Folder *SourceClassic*, so Robot *R1* is associated with User *U1*, *R2* is associated with *U2* and so on. The usernames of Robots provisioned to *U1*, *U2*, *U3* and *U4* match the usernames used by Robots *R1*, *R2*, *R3* and *R4*.
- The Classic Folder, *SourceClassic*, is mapped to a root Modern Folder named *TargetModern*. Each Environment of *SourceClassic* is mapped to a subfolder of *TargetModern*.
- For each Process in *SourceClassic*'s Environments, a corresponding Process is created in the subfolder that represents that Environment. For instance, since Environment *E1* is mapped to subfolder *TargetModern/SubModern1*, Processes *P1* and *P2* are created in *TargetModern/SubModern1*.
- Assets and Queues are created in the same subfolders of Processes that utilize them, such as Asset *A4* created in *TargetModern/SubModern2* and Queue *Q3* created in *TargetModern/SubModern3*. However, to avoid duplicated Assets and Queues in different subfolders, entities shared by multiple Processes are included in an auxiliary subfolder, *TargetModern/SubModernAuxiliary*, such as Assets *A1*, *A2*, *A3*, and Queue *Q1*.

- Time Triggers are created in the same subfolders of Processes that utilize them. For example, Time Trigger *T3* is created in *TargetModern/SubModern2* because this is the subfolder that contains Process *P3*. In the case of Queue Triggers, it is necessary that the Trigger is created in the same subfolder as the Queue and as the Process to be triggered.
- Users that represent Robots are added to subfolders containing Processes that they should execute and to subfolders containing entities used by those Processes. For example, Users *U1* and *U2* are added to *TargetModern/SubModern1* because Processes *P1* and *P2* are in *TargetModern/SubModern1*. They are also assigned to *TargetModern/SubModernAuxiliary* since they need access to Assets and Queues defined in *TargetModern/SubModernAuxiliary*: *A1*, *A2*, *A3* and *Q1*. On the other hand, User *U4* can be assigned only to *TargetModern/SubModern3* because the Processes it should execute (i.e., *P5*) and entities related to such Processes (i.e., *A5* and *Q3*) are all in *TargetModern/SubModern3*.

Figure 20 - Configuration of Entities in Modern Folders After Migration

After Users are created at tenant level, the implementation of the described migration plan can be done according to 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 21).

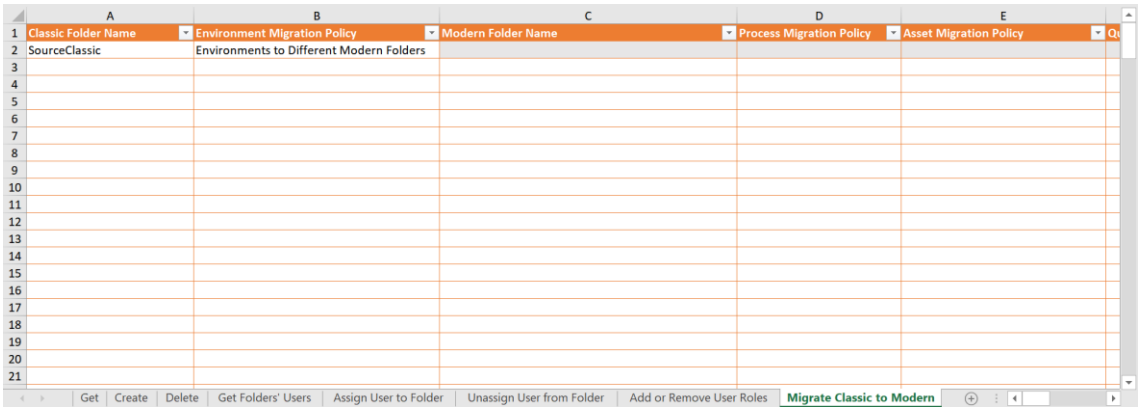


	A	B	C	D	E	
1	Type	Folder Name	Roles Assignment Model	Parent Folder Name	ID	Result
2	Modern	TargetModern	Custom Roles			
3	Modern	SubModern1	Custom Roles	TargetModern		
4	Modern	SubModern2	Custom Roles	TargetModern		
5	Modern	SubModern3	Custom Roles	TargetModern		
6	Modern	SubModernAuxiliary	Custom Roles	TargetModern		

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 - Creation of Modern Folders

2. In the **Migrate Classic to Modern** sheet of **Folders.xlsx**, input the name of the Classic Folder to be migrated, *SourceClassic*, and choose *Environments to Different Modern Folders* as Environment Migration Policy (Figure 22). 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.

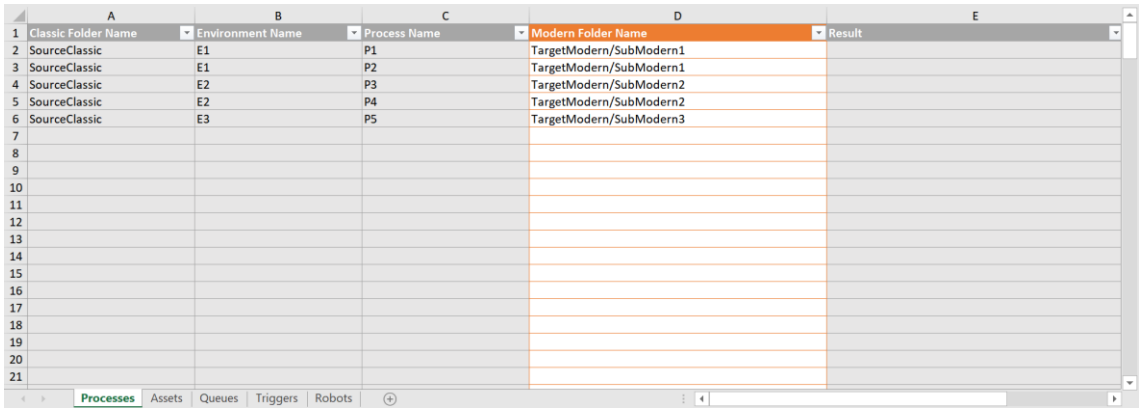


	A	B	C	D	E
1	Classic Folder Name	Environment Migration Policy	Modern Folder Name	Process Migration Policy	Asset Migration Policy
2	SourceClassic	Environments to Different Modern Folders			

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

Figure 22 - Migrate Classic Modern Sheet

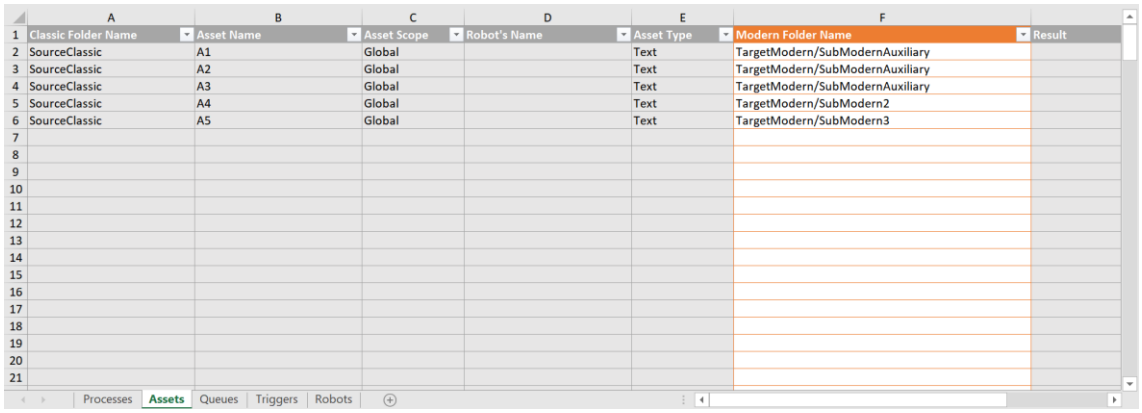
3. After *SourceClassic's* migration workbook is created and presented, input the names of the subfolders to which entities should be copied, as shown in Figure 23, Figure 24, Figure 25, Figure 26 and Figure 27. Note that, in Figure 27, the rows for Robots *R1*, *R2* and *R3* were manually duplicated to account for the assignment of the corresponding Users to multiple subfolders.
4. Start the migration by confirming it using the Migration Panel (Figure 28). Orchestrator Manager will 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	TargetModern/SubModern1	
3	SourceClassic	E1	P2	TargetModern/SubModern1	
4	SourceClassic	E2	P3	TargetModern/SubModern2	
5	SourceClassic	E2	P4	TargetModern/SubModern2	
6	SourceClassic	E3	P5	TargetModern/SubModern3	

Processes | Assets | Queues | Triggers | Robots

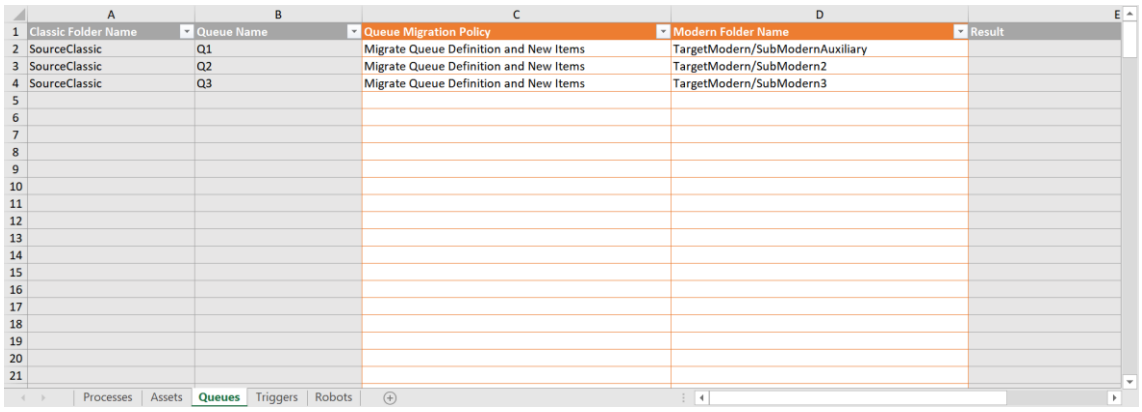
Figure 23 - Migration of Processes to Modern Folders



	A	B	C	D	E	F	
1	Classic Folder Name	Asset Name	Asset Scope	Robot's Name	Asset Type	Modern Folder Name	Result
2	SourceClassic	A1	Global		Text	TargetModern/SubModernAuxiliary	
3	SourceClassic	A2	Global		Text	TargetModern/SubModernAuxiliary	
4	SourceClassic	A3	Global		Text	TargetModern/SubModernAuxiliary	
5	SourceClassic	A4	Global		Text	TargetModern/SubModern2	
6	SourceClassic	A5	Global		Text	TargetModern/SubModern3	

Processes | Assets | Queues | Triggers | Robots

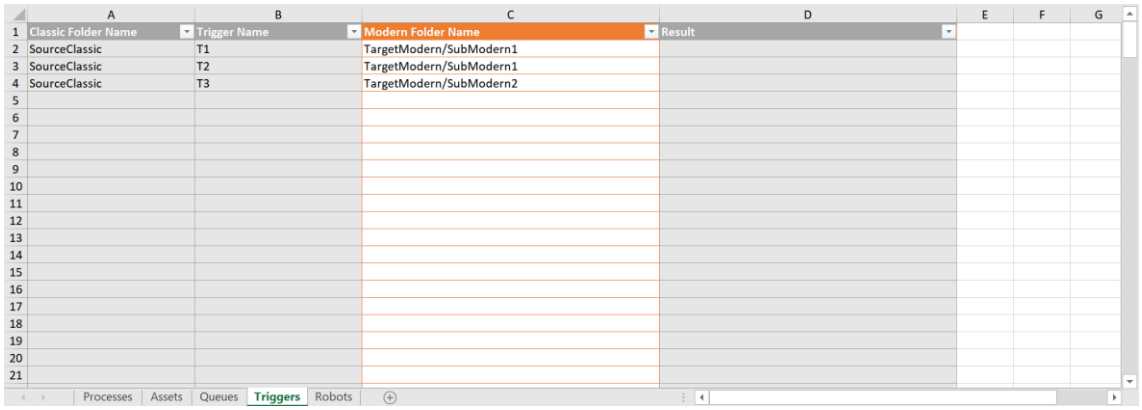
Figure 24 - Migration of Assets to Modern Folders



	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	TargetModern/SubModernAuxiliary	
3	SourceClassic	Q2	Migrate Queue Definition and New Items	TargetModern/SubModern2	
4	SourceClassic	Q3	Migrate Queue Definition and New Items	TargetModern/SubModern3	

Processes | Assets | Queues | Triggers | Robots

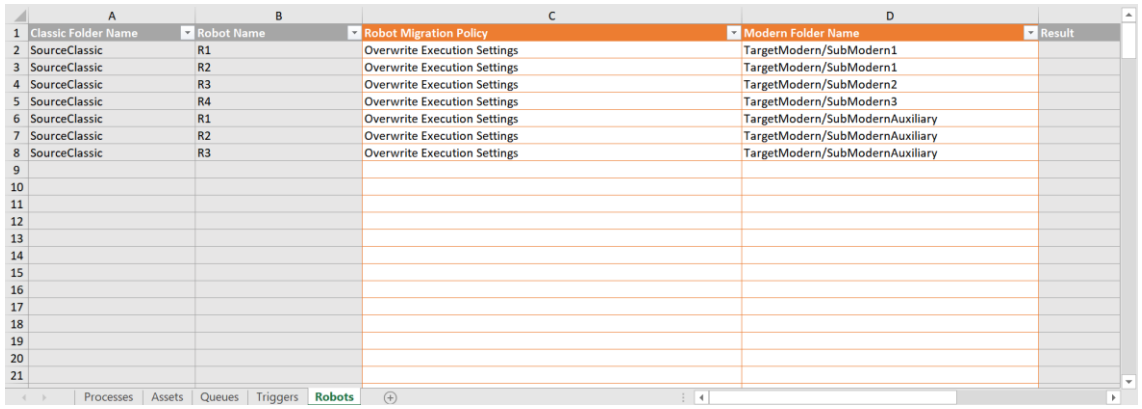
Figure 25 - Migration of Queues to Modern Folders



	A	B	C	D	E	F	G
1	Classic Folder Name	Trigger Name	Modern Folder Name	Result			
2	SourceClassic	T1	TargetModern/SubModern1				
3	SourceClassic	T2	TargetModern/SubModern1				
4	SourceClassic	T3	TargetModern/SubModern2				
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							

Processes | Assets | Queues | Triggers | Robots

Figure 26 - 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	TargetModern/SubModern1	
3	SourceClassic	R2	Overwrite Execution Settings	TargetModern/SubModern1	
4	SourceClassic	R3	Overwrite Execution Settings	TargetModern/SubModern2	
5	SourceClassic	R4	Overwrite Execution Settings	TargetModern/SubModern3	
6	SourceClassic	R1	Overwrite Execution Settings	TargetModern/SubModernAuxiliary	
7	SourceClassic	R2	Overwrite Execution Settings	TargetModern/SubModernAuxiliary	
8	SourceClassic	R3	Overwrite Execution Settings	TargetModern/SubModernAuxiliary	
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

Processes | Assets | Queues | Triggers | Robots

Figure 27 - Migration of Robots to Modern Folders

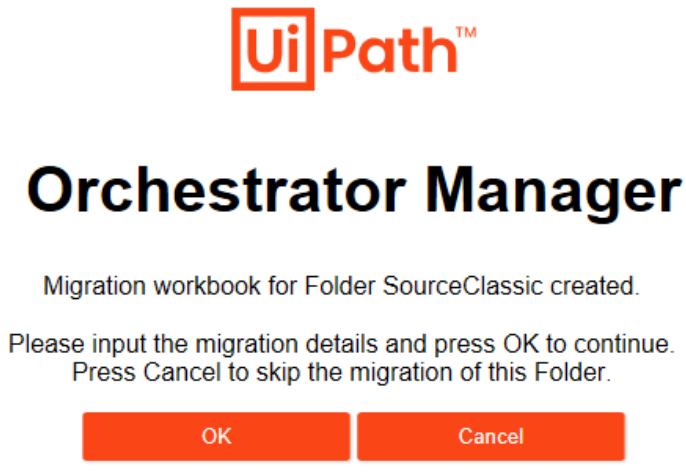


Figure 28 - Migration Panel for SourceClassic

The migration of Unattended Robots requires one more manual step, which is the assignment of Machine Templates to Modern Folders. For Processes to be executed by an Unattended Robot, there must be a Machine Template with available runtimes in the same Modern Folder as the Process itself. A Machine Template can be assigned to a Folder via Orchestrator's web interface (Figure 29).

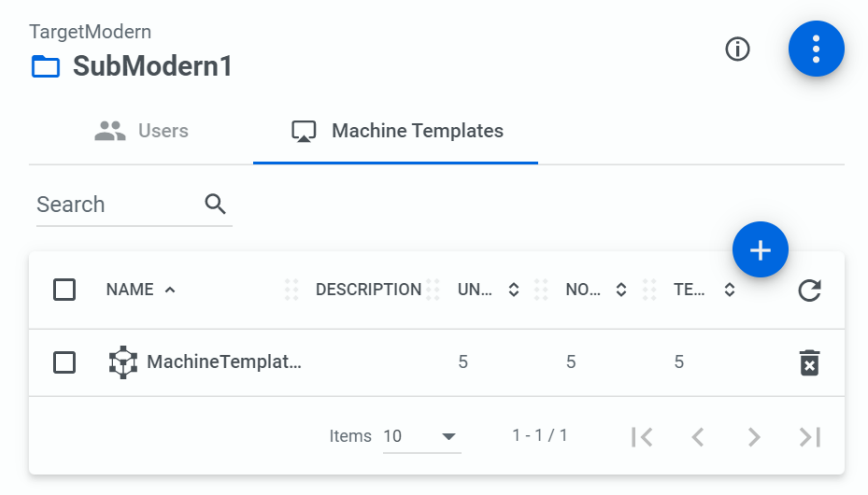


Figure 29 - Assignment of Machine Template to Modern Folder

Finally, note that some workflows may need to be updated after the migration to reflect changes in Folder structure. In particular, for Processes to access Assets and Queues from Modern Folders different than its own, 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[17]. When doing so, it is possible to specify absolute and relative paths for Modern 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 *TargetModern/SubModernAuxiliary* as value, indicating it refers to an Asset in the *TargetModern/SubModernAuxiliary* subfolder. The operation **Get Referenced Entities** on the Package entity can facilitate this step by listing what workflows in a given Package make references to what Assets and Queues.

More details about migration of entities from Classic to Modern Folders can be found at Orchestrator's online documentation[18].

---

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

[18] 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, Packages, Libraries and Triggers are currently not supported.
- Package feeds per Folder are currently 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 Marketplace[19].

This tool is offered under the 2.0 version of the Apache License[20], and support is provided on a best-effort basis.

---

[19] https://marketplace.uipath.com/listings/orchestrator-manager

[20] http://www.apache.org/licenses/