# eRegistry Project User Guide

Date: 20 Nov 2023

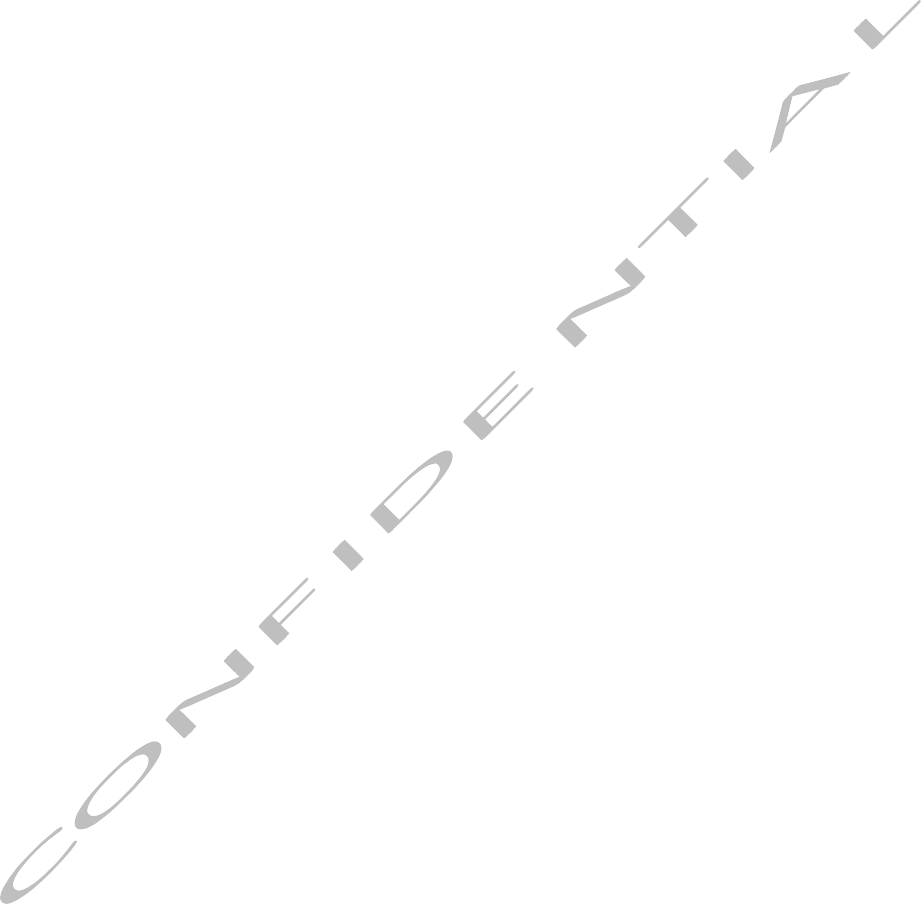
8 Mar 2024(Latest Edit)

## Project Development of Robotic Process Automation (RPA) to steamline NYC eRegistry Processes

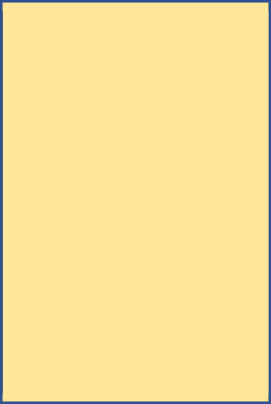
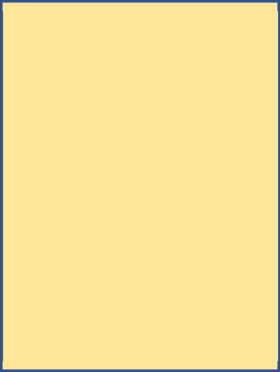
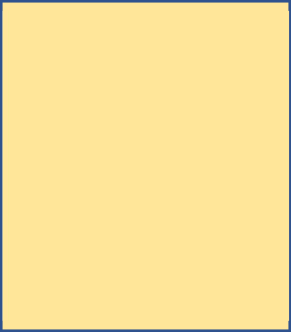
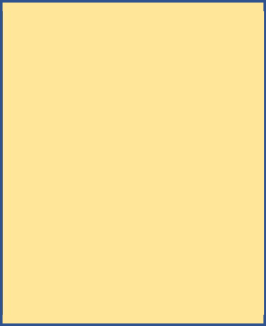
## Overview

The NYC transformation office supports NYC in the digitalisation and transformation roadmap. One of the painpoints that they have identified was the mundane processes in the e-filing and tracking of the documents created by end users in their SharePoint workspace.

Every month, many files are created by business users in NYC. These files come in various sizes, file extensions and format. The e-registry record manager will need to track them and ensure that they are not videos or larger than a certain file size. These processes are tedious and time consuming.



The **High Level As Is (Current)** e-filing and registry process is as follows:



End user drops file into DCS Workspace folders (For e-Filing)

Record manager will check through to ensure no duplicates, and ensure that each file size is <100mb, and the file format are accepted by the MCCY’s e-registry system.

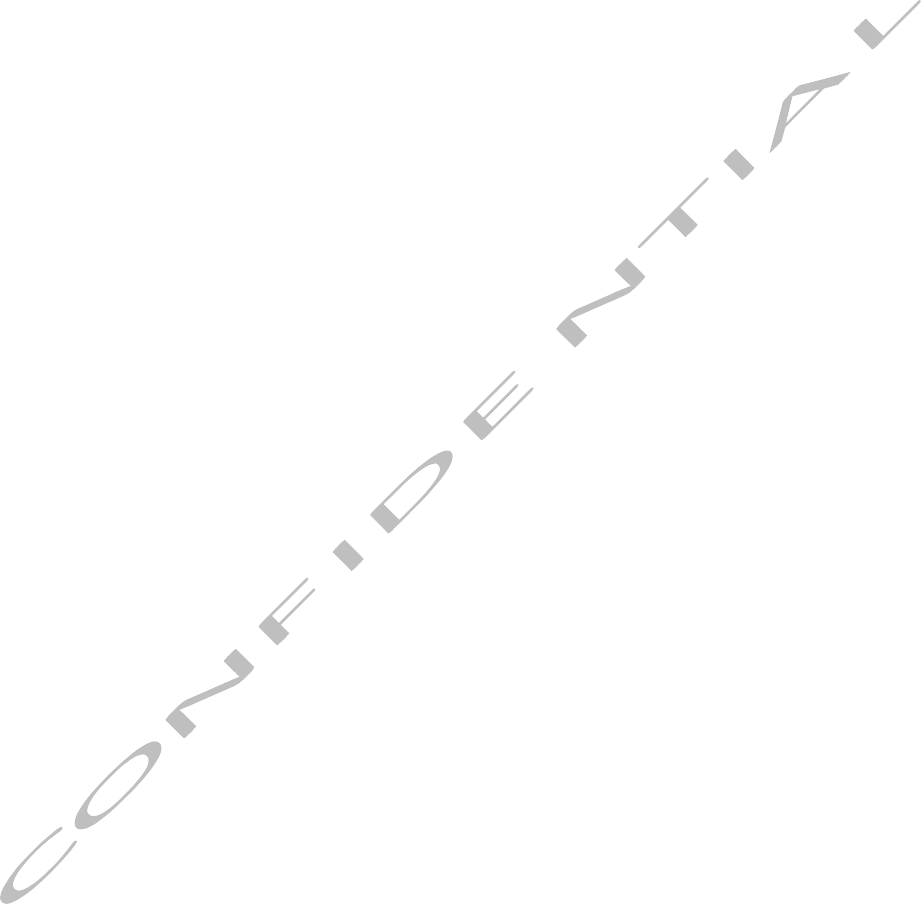
Record manager goes into the MCCY e-registry system to manually load all the files from the DCS workspace into the e-registry

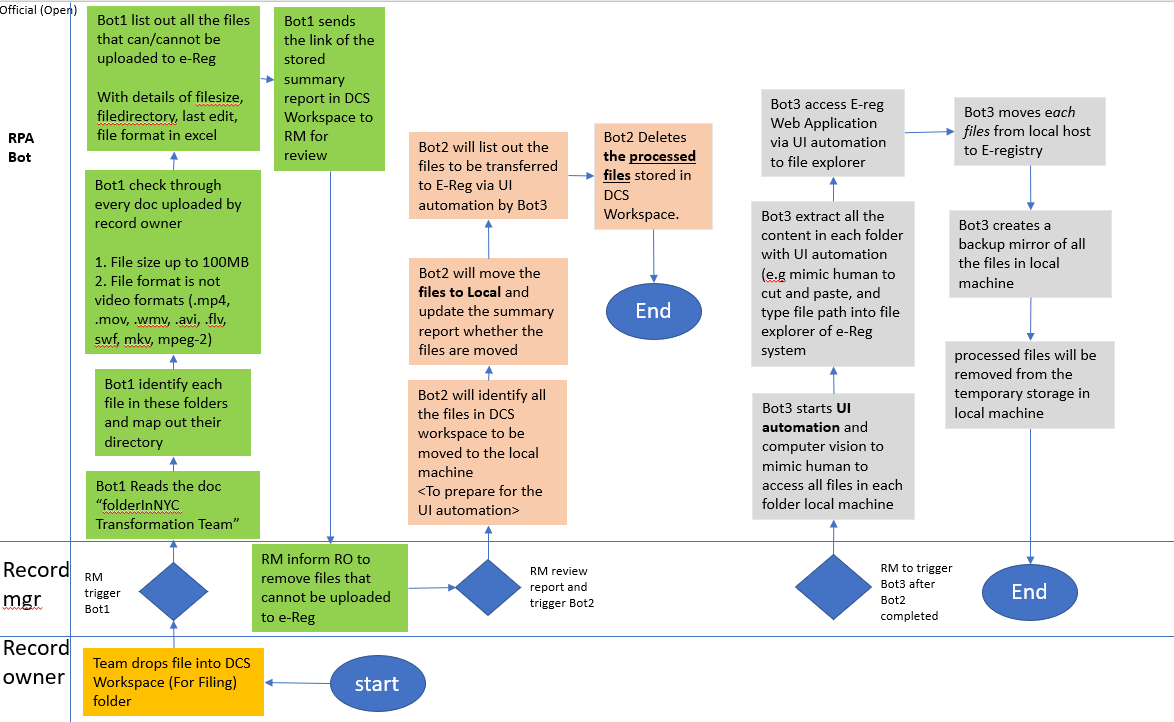
They will then manually remove all the successfully loaded files from the DCS workspace

To facilitate the above requirements and to automate some of the mundane processes, there will be a need to develop and implement **3 RPA bots** to manage the TO-BE e-registry processes as shown in the following diagram.

## Overview of TO-BE Processes

This flowchart shows the **TO-BE processes overview**:





## Tracking and Summarising Bot 1

Bot 1 is an Attended Bot which is known as RPA tool that is used by human users to assist with their day-to-day tasks. The purpose of this RPA Bot 1 is to automate the process of summarising and tracking of all the files in all the folders and sub-folders within the specified DCS workspace (SharePoint) used by NYC.

This is to ensure that we **move all the files** deposited in DCS workspace (cloud) by business users to the **local machine** and to check with the list of all existing files in MCCY’s eRegistry system. Bot1 will then list out all the files that can/cannot be uploaded to the e-Registry system with details of filesize, filedirectory, last edit, file format in the excel known as “fileAddToRegistry.xlsx”.

The DCS workspace (Cloud), NYCLocalDrive as well as eRegistry (Cloud) all function hand in hand with the excel spreadsheet known as filesAddToRegistry.xlsx and ConfigFile.xlsx. The bots are designed to be run in a specific manner and that there cannot be any skipping of steps within the operations between bot 1 and bot 2 as well as what folders are being mentioned and referenced which allows for a more dynamic and robust bot design such as allowing users to simply enter the final location etc into the excel sheet without needing to enter UIPath and change anything, at the same time by making use of the excel sheet. This allows for the use of variables to hold values allowing for easier edits. Due to this it Is very important to ensure that the folders that reside within DCS workspace mimics that of the NYCLocalDrive as well as the eRegistry. This is very important as if the eRegistry and NYCLocalDrive are not the same, this will leave the path issues when the bot attempts to run.

Below is a copy of the ExcelSheet stated previously that is provided at ur convenience which u can download(Do take note that it currently contains no information so please run bot 1 first before doing anything). There is also a copy of the ConfigFile below which is already filled up, the settings which contains the file path may need to be changed based on the pc/laptop that u are using. Change it only when necessary.



Bot1 will update 3 out of the following 6 tabs in the overall summary excel spread sheet.

The following Excel spreadsheet that is mentioned is very powerful as it governs the way bot 1 and bot 2 work with each other. What this means is that e.g Bot1 when ran logs files into TAB1 and anything IN TAB1 will only be moved.

|  |  |
| --- | --- |
|  | **Name of the Excel Tabs in the spread sheet** |
| Tab 1 | ToBeTransferredToERegistry |
| Tab 2 | NotToBeTransferredToERegistry |
| Tab 3 | SummaryOfAllFiles |
| Tab 4 | FilesMovedToLocal |
| Tab 5 | LocalFilesNotMovedtoReg |
| Tab 6 | Log |

These three tabs are important as it lets the user know the files that are being identified to be moved to e-registry in any particular month, and the corresponding files that should not be moved due to the conditions set (e.g file size >100mb and wrong file type). It also lets the user know what how many files will be transferred as well as the number of files that will not be transferred due to the certain conditions set.

The NYC record manager will then run the RPA Bot 1 once a month **as shown in** the flowchart of the **TO-BE process.**

The following shows the data in one of the tabs in the summary excel sheet. This tab is also labelled as “ToBeTransferredToERegistry” to enable user to know the list of files that will be transferred to e-Registry in each month.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **FileName** | To provide the file name of each document identified by Bot1 | pic2.jpeg | RPA Bot1 |
| **FileSize (kb)** | To provide the file size information to the users | 46.01 | RPA Bot1 |
| **File** | To provide the full file path of the file | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA\pic2.jpeg | RPA Bot1 |
| **LastEdit** | To show the last edited date/Date when RPA Bot 1 was used to transfer files over to the local machine | 09/12/2023 14:37:38 | RPA Bot1 |
| **FileExtension** | To provide details on the file extension for the Bot to check on the file type that are allowed to be uploaded into the e-Registry system | .jpeg | RPA Bot1 |
| **FileDirectory** | To provide details on the folder directory to be used by other Bots | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA | RPA Bot1 |

As Bot1 automates and runs through the set of instructions, it will also find out the list of files that **should not** be transferred to eRegistry like >100mb, wrong file type (.mp4) or duplicates. The details of the tab “NotToBeTransferredToERegistry” are shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **FileName** | To provide the file name of each document identified by Bot1 | Hosanna.mp4 | RPA Bot1 |
| **FileSize (kb)** | To provide the file size information to the users | 156914.5 | RPA Bot1 |
| **File** | To provide the full file path of the file | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA\Hosanna.mp4 | RPA Bot1 |
| **LastEdit** | To show the last edited date/Date of RPA Bot 1s last use | 09/12/2023 14:37:38 | RPA Bot1 |
| **FileExtension** | To provide details on the file extension for the Bot to check on the file type that are allowed to be uploaded into the e-Registry system | .mp4 | RPA Bot1 |
| **FileDirectory** | To provide details on the folder directory to be used by other Bots | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA | RPA Bot1 |

When RPA Bot 1 runs, it will also collate the **total number of files** to be transferred along with how many files that will not be transferred. These are all done within the **Log Tab** in the “filesaddtoregistry.xlsx” excel sheet. An example of this can be seen below as reference.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **LogMessage** | To show the amount of files that are to be transferred which are located in “ToBeTransferredToEregistry” as well as how many files are not to be transferred | Bot 1 completed run successfully. 95 files will be transferred and 1 files will not be transferred. | RPA Bot1 |
| **Remarks** | To show the last edited date/When RPA Bot 1 was last clicked | 03/06/2024 17:11:04 | RPA Bot1 |

Bot1 will use some of the settings the Config.xlsx excel spreadsheet to know which folder to crawl for file information and details.

|  |  |
| --- | --- |
|  | **Name of the Excel Tabs in the spread sheet** |
| Tab 1 | Settings |
| Tab 2 | List of Extensions |
| Tab 3 | Constants |

This ConfigFile.xlsx is important as the settings tab contains the file path that Bot1 uses in order function as well as contains the List of Extensions tab which determines which files are supposed to be added to “NotToBeTransferredToERegistry” it also contains the Constants tab which is used to obtain the value of log messages.

The ConfigFile.xlsx can be found within the data folder stored under RPA Bot1 main folder as a subfolder. It contains 3 tabs namely Settings, ListOfExtentions and Constants, ListOfExtentions contains all the extensions that are allowed to be transferred to “ToBeTransferredToERegistry”. Any edits to the extensions will directly correspond to what RPA Bot 1 does when it sees the extension added to the DCS workspace be it transferring to “ToBeTransferredToERegistry” or to “NotToBeTransferredToERegistry”.

* 1. **file transfer from cloud to local machine for Bot2**

Bo2 is a Attended Bot which is an RPA tool that is used to assist human users and in this case this bot is used to assist with the users file transfer. A comprehensive list of all existing files in a particular e-registry folders will need to be prepared in advance. This will be required by the Bot2 as it will update the summary table of all files that has been moved to the local machine. The tabs (highlighted in blue) below will be updated by the bot2.

|  |  |
| --- | --- |
|  | **Name of the Excel Tabs in the spread sheet** |
| Tab 1 | ToBeTransferredToERegistry |
| Tab 2 | NotToBeTransferredToERegistry |
| Tab 3 | SummaryOfAllFiles |
| Tab 4 | FilesMovedToLocal |
| Tab 5 | LocalFilesNotMovedtoReg |
| Tab 6 | Log |

The details of the “SummaryOfAllFiles” tab is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **FileName** | To provide the file name of each document identified by Bot1 | pic2.jpeg | RPA Bot2 or past info provided before we go “Live” |
| **FileSize (kb)** | To provide the file size information to the users | 46.01 | RPA Bot2 or past info provided before we go “Live” |
| **File** | To provide the full file path of the file | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA\pic2.jpeg | RPA Bot2 or past info provided before we go “Live” |
| **LastEdit** | To show the last time that Bot2 was ran. | 09/12/2023 14:37:38 | RPA Bot2 or past info provided before we go “Live” |
| **FileExtension** | To provide details on the file extension for the Bot to check on the file type that are allowed to be uploaded into the e-Registry system | .jpeg | RPA Bot2 or past info provided before we go “Live” |
| **FileDirectory** | To provide details on the folder directory to be used by other Bots | C:\Users\alanchow\OneDrive - Nanyang Polytechnic\DCS workspace\NYC-Core Business\RPA | RPA Bot2 or past info provided before we go “Live” |

After Bot2 has completed its run, it will also show a list of all the “**FilesMovedToLocal**” as shown in the table below. This is also to ensure that we know all the files in the “ToBeTransferredToERegistry” are being processed or not.

If they are all transferred, all the file record in ToBeTransferredToERegistry will now be in the “**FilesMovedToLocal**”

The objective of moving to the local machine is to overcome some of these challenges of a direct FTP from DCS workspace to eRegistry. Hence, the files moved to the local will enable Bot3 to complete the final transfer via computer vision (CV) and UI automation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **FileName** | To provide the file name of each document identified by Bot1 | pic2.jpeg | RPA Bot2 |
| **FileSize (kb)** | To provide the file size information to the users | 46.01 | RPA Bot2 |
| **File** | To provide the full file path of the file | C:\Users\alanchow\ \pic2.jpeg | RPA Bot2 |
| **LastEdit** | To show the last edited date | 09/12/2023 14:37:38 | RPA Bot2 |
| **FileExtension** | To provide details on the file extension for the Bot to check on the file type that are allowed to be uploaded into the e-Registry system | .jpeg | RPA Bot2 |
| **FileDirectory** | To provide details on the folder directory to be used by other Bots | C:\Users\alanchow\ | RPA Bot2 |
| **ProcessedFlag** | To provide the details on whether it has been processed or moved.  Hence if the file is processed and its name is in this list, Bot1 when run in the next month will know that the new files are processed | 1 or 0 | RPA Bot2 |
| **Remarks** | To provide the details of any exception |  | RPA Bot2 |

If bot2 is run successfully, all the files within DCS workspace will be moved to NYCLocalDrive and they will be removed from DCS workspace. It will also generate a Log similar to Bot1 under the “Log” Tab “filesaddtoregistry.xlsx” excel sheet. An example of the log message can be seen below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **LogMessage** | To show the amount of files that are not to be transferred referenced by the tab “FilesMovedToLocal” and only those that have ProcessedFlag of “1” | Bot 2 completed run successfully. 3 files will be transferred | RPA Bot2 |
| **Remarks** | To show the last edited date/When RPA Bot 2 was last clicked | 03/06/2024 17:11:04 | RPA Bot2 |

## UI Automation & computer vision for Bot3

Bot3 is known as a Attended Bot which is a type of RPA tool that is designed to work alongside human users by assisting them with their day to day task. The purpose of this Bot 3 is to mimic human and automate the steps of the manual task of the officer in charge (e.g mimic human actions like copying information, pasting information, and typing all the unique file paths into text boxes of file explorer or the e-Registry system).

Similarly like bot1, Bot3 also contains a config.xlsx which is used to simply store the file path for the FinalLocation which in this case would be the E-Registry cloud, this allows for the user to simply enter the excelsheet if they wish to modify the finalLocation without having to understand UIpath. If this config excelsheet is deleted/Misplaced, it will lead to errors as the bot would not know where to move the files under NYCLocalDrive. Below contains the Config file that was used which has been provided for your convenience. Do note that **it does not** have the finalLocation otherwise known as E-Registry configured, if you plan on using this please ensure that u setup the correct file path.



Next, Bot3 will first leverage all the information gathered by Bot1 and Bot2 and go to the specific folders in the local machine and get the corresponding files. Then, it will launch a file explorer interface. This is required for accessing MCCY’s eRegistry. The Bot3 will analyse the file location of the current files in the local machine, then key in the file path required to go into the new location in eRegistry.

The Bot3 will also automatically traverse to each sub-folder in the mapped temporary folders in the local that mirrors the DCS workspace and eRegistry, namely the **“NYC-Core Business” and the “NYC-Corporate support”**. Then, it will grab all the files and do a cut and paste into the new location. Once all the temporary folders are cleared, it will indicate that Bot3 has completed its tasks.

Lastly, the team understands that UI automation may potentially lead to timeouts due to server displaying the output slower than the Bot3’s run speed and other unique scenarios may not be easily captured. Hence, we will also explore additional features of having Bot3 to check through these local folders and identify the files not moved and log into the summary sheet tab “LocalFilesNotMovedToReg”. The following shows the details of the “LocalFilesNotMovedToReg” tab.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **FileName** | To provide the file name of each document identified by Bot1 | pic2.jpeg | RPA Bot3 |
| **File** | To provide the full file path of the file | C:\Users\alanchow\ \pic2.jpeg | RPA Bot3 |
| **FileDirectory** | To provide details on the folder directory to be used by other Bots | C:\Users\alanchow\ | RPA Bot3 |
| **Remarks** | To provide the details on whether it has been processed or moved. | The file is not moved | RPA Bot3 |

Bot3 similarly like Bot1 and Bot2 will generate Logs upon the completion of bot3s process. This will likewise be generated in “Log” Tab “filesaddtoregistry.xlsx” excel sheet. An example of the log message can be seen below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Example of the data** | **Updated by** |
| **LogMessage** | To show the amount of files that are not to be transferred referenced by the tab “LocalFilesNotMovedToReg” | Bot 3 completed run successfully. 3 files will not be transferred | RPA Bot3 |
| **Remarks** | To show the last edited date/When RPA Bot 3 was last clicked | 03/06/2024 17:11:04 | RPA Bot3 |

**Important!**

Upon seeing any document in this list, user may need to recheck if the VPN network to eRegistry is properly establish or any document has been accidentally used by other local application. Then, you may re-run Bot3 again to check for unmoved files. Once this list is empty, the move of all localfiles to eRegistry has been fully completed.

**Guide For Advance RPA administrators**

You will need to make sure that u have 4 files downloaded from the website and placed within the file location “C:\ProgramData\UiPath\Packages “ of any user’s local PC/laptop. Thi order to deploy Bot1 and Bot2 with the use of UiPath Assistant. Otherwise, this will cause u to run into errors like “dependency files do not exist”.

<https://www.nuget.org/>

And these files are:

1. runtime.linux-arm.runtime.native.System.IO.Ports
2. runtime.linux-arm64.runtime.native.System.IO.Ports
3. runtime.linux-x64.runtime.native.System.IO.Ports
4. runtime.osx-x64.runtime.native.System.IO.Ports

You can type the above details 4 files into the website (<https://www.nuget.org/>) or simply just download the 4 files below.

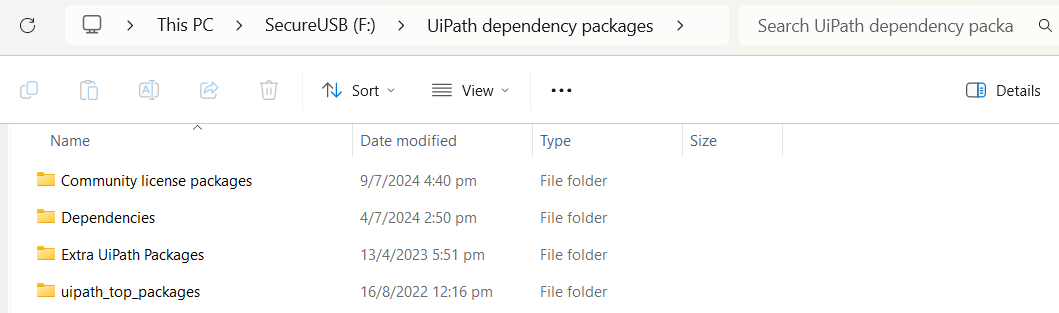


For Bot3, you will need the following 33 files to be put in “C:\ProgramData\UiPath\Packages “ of any user’s local PC/laptop. Alternatively, you can also go to <https://www.nuget.org/> to find them.

1. **runtime.debian.8-x64.runtime.native.System.Security.Cryptography.OpenSsl**
2. **runtime.fedora.23-x64.runtime.native.System.Security.Cryptography.OpenSsl**
3. **runtime.fedora.24-x64.runtime.native.System.Security.Cryptography.OpenSsl**
4. **runtime.opensuse.13.2-x64.runtime.native.System.Security.Cryptography.OpenSsl**
5. **runtime.opensuse.42.1-x64.runtime.native.System.Security.Cryptography.OpenSsl**
6. **runtime.osx.10.10-x64.runtime.native.System.Security.Cryptography.OpenSsl**
7. **runtime.rhel.7-x64.runtime.native.System.Security.Cryptography.OpenSsl**
8. **runtime.ubuntu.14.04-x64.runtime.native.System.Security.Cryptography.OpenSsl**
9. **runtime.ubuntu.16.04-x64.runtime.native.System.Security.Cryptography.OpenSsl**
10. **runtime.ubuntu.16.10-x64.runtime.native.System.Security.Cryptography.OpenSsl**
11. **runtime.osx.10.10-x64.runtime.native.System.Security.Cryptography.Apple**
12. **runtime.any.System.Collections**
13. **runtime.win.System.Diagnostics.Debug**
14. **runtime.any.System.Diagnostics.Tracing**
15. **runtime.any.System.Globalization**
16. **runtime.any.System.IO**
17. **runtime.win.System.IO.FileSystem**
18. **runtime.win.System.Net.Primitives**
19. **runtime.any.System.Resources.ResourceManager**
20. **runtime.any.System.Runtime**
21. **runtime.win.System.Runtime.Extensions**
22. **runtime.any.System.Runtime.Handles**
23. **runtime.any.System.Runtime.InteropServices**
24. **runtime.any.System.Text.Encoding**
25. **runtime.any.System.Threading.Tasks**
26. **runtime.any.System.Reflection**
27. **runtime.any.System.Reflection.Primitives**
28. **runtime.any.System.Globalization.Calendars**
29. **System.Threading.Overlapped**
30. **System.Private.Uri**
31. **runtime.any.System.Text.Encoding.Extensions**
32. **runtime.win.Microsoft.Win32.Primitives**
33. **runtime.win7.System.Private.Uri**



In case the above still does not solve the issue or is prompting you for dependency errors, refer to the folder called “UiPath dependency packages”. This folder can be found in the deployment thumb drive, once you access the folder there should be 4 subfolders:



Which folder do I copy?

* First you look at the errors prompted in UiPath (The error will appear under the output in UiPath), it will state the files that are not found.
* From here, add in the folder(s) that you need **based on the errors prompted**, for example, if UiPath prompts for a certain missing file, look for the file in the folder by searching for the file name, if found, copy the folder to your local pc (can be anywhere as long as you link it later).

Here’s an example:

A screenshot of a computer

Description automatically generated

In this example, UiPath prompted me for Microsoft graph, so I went to search for it and found it under the “Community license packages” folder, so here I will copy the folder over to my local pc.

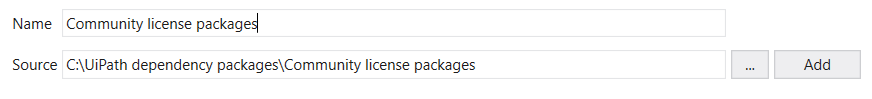
From here just link it in UiPath under packages:

A screenshot of a computer

Description automatically generated

Here, click on the “+” to add a new package and link it to the folder you copied over.

Example:



Then, check the folder you just added, this should solve the errors prompted.

A close-up of a logo

Description automatically generated

You will also need a local PC/laptop that has sufficient storage to temporarily store the files from DCS workspace. Then, you will need to always create a folder in C drive name as NYCLocalDrive. For instance, “C:\NYCLocalDrive”. It must have a mirror subfolder of all the subfolder that you want to automate. For instance, **“C:\NYCLocalDrive\NYC-Core Business” and the “C:\NYCLocalDrive\NYC-Corporate support”** and the other subfolders inside these folders.

There are also 2 files that are required for the 3 Bots to function well.

First is the **ConfigFile.xlsx**. It has 3 tabs. One of it stores the list of subfolders that the Bots will traverse to in the DCS workspace. For scalability, if there is a need to increase the number of folders, we will need to go to a tab to add the new folder name (e.g NYC-Human resource).

Subsequently, you will also need to create the additional folder in the local PC in the NYCLocalDrive. In this scenario, you will now have 3 folders namely, **“C:\NYCLocalDrive\NYC-Core Business” and the “C:\NYCLocalDrive\NYC-Corporate support”, and**  “**C:\NYCLocalDrive\NYC-Human resource”**.

It also has a tab known as ListOfExtensions which is also for RPA advance user to scale up the other types of files that we want to disallow user from putting into the eRegistry. E.g .exe or .js file.

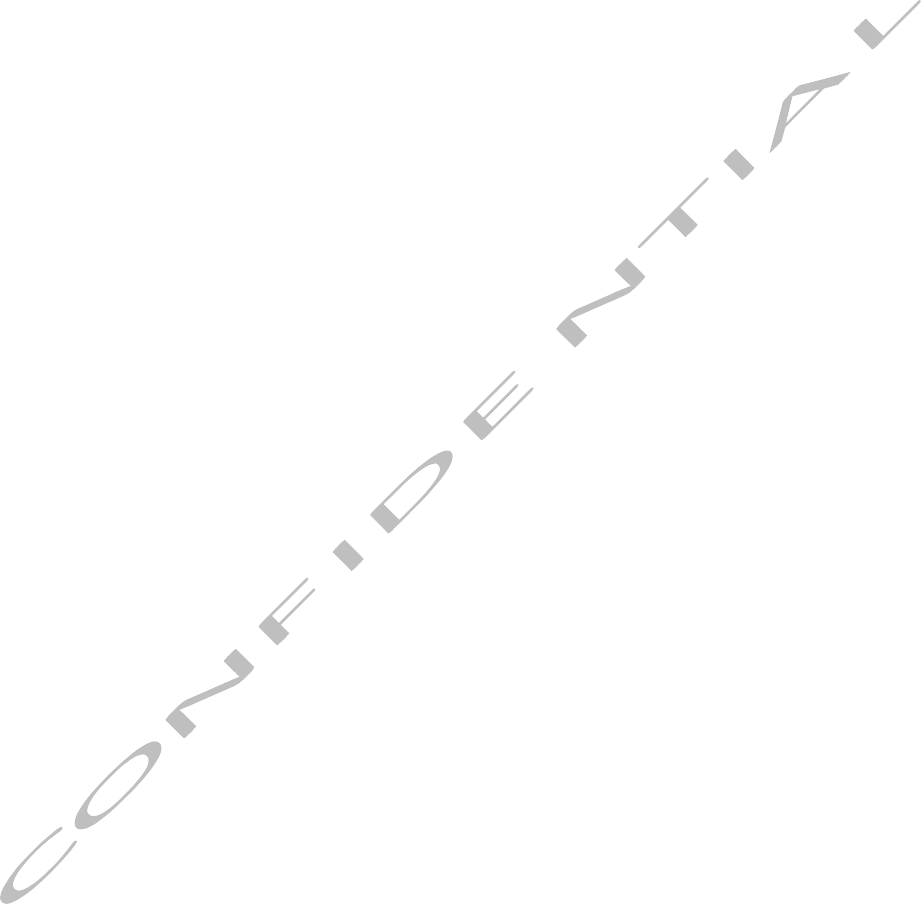
It has a final tab known as Constants which stores 6 different log messages which the user can use if they wish to change the message displayed.

The other file is the “fileAddToRegistry”, with the 6 tabs that has been use and described in the above illustrations

|  |  |
| --- | --- |
|  | **Name of the Excel Tabs in the spread sheet** |
| Tab 1 | ToBeTransferredToERegistry |
| Tab 2 | NotToBeTransferredToERegistry |
| Tab 3 | SummaryOfAllFiles |
| Tab 4 | FilesMovedToLocal |
| Tab 5 | LocalFilesNotMovedtoReg |
| Tab 6 | Log |

There is also another config.xlsx file that is used by bot3 as stated above under the bot3’s illustration and this ExcelSheet has a single tab known as settings which stores the finalLocation which in this case is E-Registry. This is used by Bot3 in order to traverse the subfolders located within the finalLocation in order to move files accurately as well as seamlessly.

## Hardware/Software Requirement for this project



This section shows the targeted platform that will be used during the development of project.

|  |  |  |
| --- | --- | --- |
| Requirement | | |
| 1 | Software | UiPath Studio Developer Edition 2022.10.5.0 UiPath Robot /Assistant licensed account |
| 2 | Hardware | Personal computer (PC) installed with Uipath software and access to all required intranet/internet systems and is safe to transfer files into and from.  The PC must also have storage space of ~500GB and is able to access rights to MCCY’s eRegistry and NYC’s SharePoint and DCS workspace. |