**IBM Econt RPA Process Administrative Reports**

за проект „Изготвяне на административни справки“

**Описание на RPA процеса**

**и**

**Ръководство за употреба**

# История и редакции на документа

**Име на RPA проекта:** IBM Econt RPA Process Administrative Reports

**Статус на документа (пр. Чернова, Финален, Издание #):** Издание 1.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Версия на документа | Дата | Изготвен от | Редактиран от | Одобрен от | Статус |
| 1.0 | 2021-04-22 | Иво Стоименов  Стефан Стефанов |  |  | Издание 1.0 |

Contents

[1. История и редакции на документа 2](#_Toc73538793)

[2. Таблици 4](#_Toc73538794)

[3. Цел и обхват на документа 4](#_Toc73538795)

[4. Описание на процеса 4](#_Toc73538796)

[4.1. Преглед 4](#_Toc73538797)

[4.2. Transaction Queue (Transaction Log) 5](#_Toc73538798)

[4.3. Външен конфигурационен файл 6](#_Toc73538799)

[4.4. Детайлно описание на процеса 6](#_Toc73538800)

[ Dispatcher 6](#_Toc73538801)

[ Processor 7](#_Toc73538802)

[4.5. Изпращане на мейли при завършване или нередност 7](#_Toc73538803)

[ Мейл за нередност 7](#_Toc73538804)

[5. Работа с робота 8](#_Toc73538805)

[5.1. Промяна на конфигурационни параметри 8](#_Toc73538806)

[5.2. Стартиране на процеса 8](#_Toc73538807)

[5.3. Проверяване на лог файловете на IBM RPA Launcher 9](#_Toc73538808)

[5.4. Публикуване на нов процес в облачното пространство 10](#_Toc73538809)

[6. Поддръжка на робота 11](#_Toc73538810)

[6.1. Поддръжка на дисковото пространство 11](#_Toc73538811)

[6.2. Поддръжка на пощенската кутия 11](#_Toc73538812)

[7. Преместване на робота на друга машина 12](#_Toc73538813)

[7.1. Операционни изисквания 12](#_Toc73538814)

[ Хардуер и Софтуер 12](#_Toc73538815)

[ https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct?deliverableId=A3386110AF3511EAB9117899910938C8 12](#_Toc73538816)

[ Мрежови 12](#_Toc73538817)

[ Достъп на Outlook до мейл сървъра за изпращане на мейли; 12](#_Toc73538818)

[ Достъп до споделени файлове GoogleSheets; 12](#_Toc73538819)

[7.2. Инсталиране на IBM RPA Studio 12](#_Toc73538820)

[ Инсталация 12](#_Toc73538821)

[ Активация 12](#_Toc73538822)

[7.3. Инсталиране и конфигуриране на приложенията 12](#_Toc73538823)

[7.4. Преместване на работните файлове 13](#_Toc73538824)

[7.5. Добавяне на Vault credentials 13](#_Toc73538825)

[9.7 Задаване на график за стартиране на процеса 15](#_Toc73538826)

# Таблици

[Таблица 1 Описание на колоните в TransactionQ файла 5](#_Toc67393210)

# Цел и обхват на документа

Документът е описание на функциите и ръководство за употреба на роботната автоматизация на процес „***Econt RPA Process Administrative Reports***“.

Процесът се изпълнява от имплементация на софтуерен робот на платформа IBM RPA Studio при Еконт .

# Описание на процеса

## Преглед

Целта на процес ***„*Econt RPA Process Administrative Reports*“*** е обработване на подадени клиентски заявки в системата Jira, където има специално отделена опашка на робота “Robot Process Automation – RPA”, валидиране на част от данните, извличане на справка от системата на Еконт, изготвяне на желаната заявка и уведомително писмо към нея и изпращане на файловете по e-mail към клиента или към съответният Reporter (при липса на клиентски e-mail).

За всяка заявка с уникално ИД от системата Jira , се изпълнява следното:

1. Влиза се в системата Jira, от където се свалят данните подадени от клиента, необходими

за извличане на заявката. Проверява се верността на ЕГН или ЕИК и обхвата на справката:

* ако има грешно въведени данни или обхвата на справката е по-голяма от наличната база данни в системата за изготвяне на справки се вписва коментар за некоректност на данните и заявката се отстранява от опашката на робота чрез “Unassign” в съответното поле.
* Ако подадените данни са верни се вписват в Queue файла на робота и файла необходим за изпълнението на скрипта при стартиране на клъстера в Azure.

1. При наличие на повече от един телефон в заявката на един клиент, се изготвят две отделни заявки за изпълнение.
2. След обработването и на последната заявка в Jira се преминава към системата Azure, където се стартира клъстера и самият скрипт за изготвяне на справките.
3. Преминава се през изтегляне на готовите справки пуснати по-рано за изготвяне.
4. Сравнява се телефонният номер на клиента от Queue файла на робота с номера на клиента вписан в заглавието на файловете от свалените изготвени справки от Azure.При съвпадение справката се обработва по съответните изисквания от клиента(включване само на необходимите листове) и филтриране на самите листове по име на клиента в съответните колони за всеки лист.
5. Данните от заявката се записват в регистъра на изготвените справки, като от там се взима поредният номер необходим за попълване на Уведомителното писмо
6. Изготвя се Уведомително писмо по шаблон със съответните данни на клиента.
7. Готовите файлове се прикачат към съответната заявка в Jira

* Ако заявката съдържа един телефон след прикачването на файловете заявката се затваря чрез натискане на бутона „за изготвяне “ и промяната му на статуса в „приключен“
* Ако заявката съдържа повече от един телефон, когато се прикачат и последните файлове тогава заявката се приключва.

1. Изпраща се e-mail с прикачените документи съответно към клиента(при попълнен e-mail) и към Reporter-a(при липсващ e-mail на клиента)

## Transaction Queue (Transaction Log)

Процесът използва външен .xlsx файл (Queue\_01.xlsx) за проследяване на прогреса с отделен статус за всяка транзакция. Една транзакция се идентифицира с „Phone“ и уникално „ID” от Jira

Queue\_01.xlsx файлът се намира в работната директория на робота.

Описание на колоните в TransactionQ файла:

|  |  |
| --- | --- |
| **Име на колона** | **Стонйности и описание** |
| TransactionID | Уникален идентификатор на транзакция |
| CreatedDate | Точния час и дата когато транзакцията е добавена в опашката |
| Status | **New**: транзакцията е добавена в опашката  **In progress**: транзакцията е взета за обработка  **Success**: транзакцията е успешно обработена  **Failed**: транзакцията е завършила с неуспех |
| FailureType | **BE**: Business exception – когато е установена очаквана грешка, при която не може да се продължи поради бизнес логиката  **SysError**: System exception – когато е установена неочаквана грешка. При такъв тип грешка транзакцията може да бъде взета отново за повторна обработка. |
| Result | Описание на резултата от обработката на транзакцията  Ако **Failure Type** **=** **BE** – Причината за грешката  Ако **Failure Type** **=** **SysError** – Системното съобщение за грешка |
| History | Записват се предишните резултати от всеки опит да се изпълни транзакцията с дата и час |
| Retries | Колко пъти е взета транзакцията за обработка, као е имало системна грешка |
| MainRunTime | Часът и датата когато е стартирал процеса, който е обработил транзакцията |
| StartTime | Часът и датата когато е стартирала обработката на транзакцията |
| EndTime | Часът и датата когато е приключила обработката на транзакцията |
| Stage | Нивото на изпълнение на транзакцията |
| ID | Идентификационен номер на заявката взет от JIRA |
| CustomerMail | Клиентски e-mail от заявката в JIRA |
| Address | Клиентски адрес подаден в заявката |
| Phone | Клиентски телефон подаден в заявката(9 цифрен обработен) |
| SubjectReference | Вид на използваната услуга(Справки за куриерски и пощенски услуги) |
| CustomerName | Името на клиента подадено в заявката |
| StartDate | Дата за начало на периода за справката |
| EndDate | Дата за край на периода за справката |
| EGN | ЕГН или ЕИК подаден в заявката |
| ConcatSpravPart | Тип желани справки от клиента  (Пр: Изпратени преводи;Получени преводи;Изпратени пратки;Получени пратки;Получени наложени платежи;Заплатени наложени платежи;) |
| EGNVerification | Дали въведената стойност е ЕГН или ЕИК |
| PhoneCounter | Броя на телефоните в заявката |
| Reporter | e-mail на Reporter-a |
| ProcessPassed | Статус на преминаване на трансакцията |
| FullPhone | Клиентски телефонен номер необработен |

Таблица 1 Описание на колоните в TransactionQ файла

## Външен конфигурационен файл

Процесът използва външен .xlsx файл за конфигуриране на променливи параметри за работа.

Конфигурационният файл се намира на определена локация предварително кодирана в робота.

Конфигурационният файл позволява да се променят параметри като:

* Път към работната директория на робота;
* Име на Credential записът в Vault Credential Manager или Clould System -> menu Credentials, използван за логване в Jira, Azure, GoogleSheets;
* Име на Outlook Profile за изпращане на мейли;
* Мейл адрес, на който да се изпращат резултатите на процеса;
* Други.

Описанието на параметрите е написано срещу всеки в колона „Description“.

Потребителят може да променя стойността на параметър когато роботът не работи. Само параметрите в лист „**Environment**“ са позволени за промяна от потребител. Другите листи не са защитени, но са за употреба от разработчика.

За да промени стойност на параметър, потребителят трябва да промени стойността в колона D „Prod environment“ срещу името на параметъра в колона „B“ в лист „**Environment**“.

## Детайлно описание на процеса

Процесът е разделен на 2 под-процеса – Dispatcher и Processor.

### Dispatcher

При стартиране на робота първо се извършва логване в системата Azure, където се проверява дали клъстера е активен или не. При активен клъстер робота приключва работата си. Ако клъстера е неактивен се влиза в Azure системата и се сваля файла необходим за изпълнение на скрипта за извличане на заявките. Свалят се и готови справки, ако са налични в Azure Storage-a.

Влиза се в системата Jira, от където се свалят данните подадени от клиента, необходими за извличане на заявката. Проверява се верността на ЕГН или ЕИК и обхвата на справката:

* ако има грешно въведени данни или обхвата на справката е по-голяма от наличната база данни в системата за изготвяне на справки, се вписва коментар за некоректност на данните и заявката се отстранява от опашката на робота чрез “Unassign” в съответното поле.
* ако подадените данни са верни те се вписват в Queue\_01 файлът на робота и файлът необходим за изпълнението на скрипта при стартиране на клъстера в AzureDatabricks.

При наличие на повече от един телефон в заявката на един клиент, се изготвят две отделни заявки за изпълнение.

Всички подадени заявки се проверяват да не се дублират с предходно обработени или чакащи за обработване и се записват в опашката за изпълнение на робота във файл (Queue\_01.xlsx).

След обработването и на последната заявка в Jira се преминава към системата Azure, където се стартира клъстера и самият скрипт за изготвяне на справките.

Сваля се файла от GoogleSheets необходим за извличане на поредният изходящ номер за поредната трансакция.

Това е диспачване на задачите за робота.

### Processor

Този под-процес взима всеки запис от Queue\_01-файла и изпълнява различни действия по логиката на робота.

1. Прави се проверка за съвпадение на стартова дата, крайна дата, име и телефонният номер на клиента от Queue\_01.xlsx и заглавията на готовите рапорти от AzureDatabricks. При съвпадение се обработва рапорта по подадените от клиента условия и се създава .xlsx файл готов за експортиране към клиента.
2. Влиза се в GoogleSheets файла и там се попълва поредният изходящ номер, името на клиента, датата, вида на справката и уникалното ID от Jira системата.
3. Попълва се Уведомителното писмо
4. Прикачват се изготвените файлове в Jira и ако прикачванията съответстват на броя на телефонните номера в подадената заявка тя се приключва и затваря.

## Изпращане на мейли при завършване или нередност

Процесът изпраща мейли през Outlook използвайки инсталирания профил на робота. Профилът от който да се изпращат мейли може да се зададе в конфигурационния файл, а името и паролата за профила се съхраняват или в облака или във Vault Credential Manager на съответната машина.

### Мейл за нередност

В случай че роботът установи нередност, при която не може да продължи обработката, се изпраща мейл с описание на проблема, ако има обработени записи.

Файлова структура

Роботът използва собствена работна директория D:\RPA ServiceCentrix\IBM FEcont\_RPA\_Process\_Administrative\_Reports.

За целите на описанието ще наричаме основната работна директория “**DataPath**”.

Структурата и съдържанието на директориите в текущата конфигурация са описани долу:

[**DataPath**]

Съдържа конфигурационния файл, образци за създаване на мейли и файлове, Transaction Log и други.

| -------> [**Exceptions\_Screenshots**]

Съдържа снимки на текущото състояние на интерфейса при установена нередност (exception).

| -------> [**Temp**]

Съдържа временни работни файлове необходими за обработка на данни.

| -------> [**TemplatesFolder**]

Съдържа образци за изпращане на мейли и други необходими за създаване на файлове.

# Работа с робота

## Промяна на конфигурационни параметри

Параметрите на процеса се задават в конфигурационния файл „*Config.xlsx*“, който по подразбиране се намира в [**DataPath**] на работната станция на робота (стойността на [**DataPath**] е записана в конфигурационния файл).

Стойностите на параметрите могат да се променят когато роботът не работи.

Файлът съдържа:

* Лист „**Environment**“ – в него се задават конфигурационните параметри за различните операционни среди:
  + Prod environment
  + Test environment
* Лист “**QueueDefinition**” задава структурата на Transaction Log файла.
* Другите листи не е необходимо да се променят от оператор.

## Стартиране на процеса

Нормално процесът стартира по график конфигуриран в [https://eu1app.wdgautomation.com/](https://eu1app.wdgautomation.com/%20%20%20%20%20%20%20) ----> Scripts ---->  за съответният процес ---> 

Операторът също може да стартира процеса от IBM RPA Launcher, който се намира в Start Menu на компютъра или на Desktop-а на машината, при условие, че е налична предварителна конфигурация от https://eu1app.wdgautomation.com/ ----> меню Launchers.

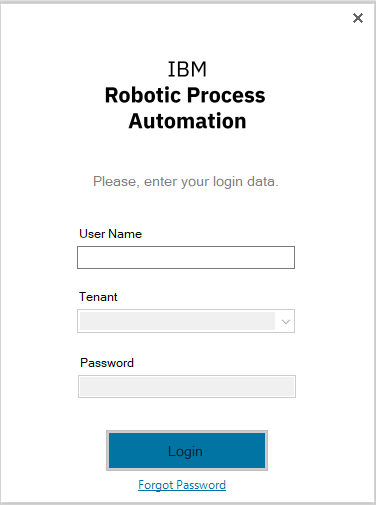
За да стартирате процес:

1. На Работния плот е поставен пряк път, от който може да се стартира IBM RPA Launcher.



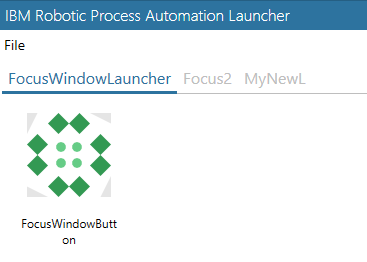
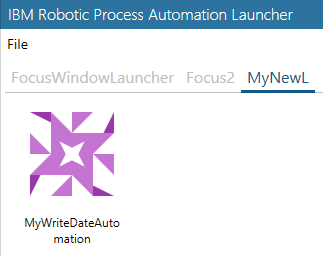
Фигура 1 Икона на IBM RPA Launcher

1. Може преди да се стартира IBM RPA Launcher-а да поиска влизане в системата със съответните Credentials



Фигура 2 IBM RPA Launcher прозорец за влизане в системата

1. Намиране на съответният таб за процеса в Launcher-а

-------->

1. Стартиране на процеса от съответната икона



## Проверяване на лог файловете на IBM RPA Launcher

UiPath Robot генерира лог файлове, в които се записват резултати от изпълнението. Стандартният път към логовете на UiPath Robot е „C:\Users\RPAadmin2\AppData\Local\IBM Robotic Process Automation\Studio.log“

## Публикуване на нов процес в облачното пространство

За да бъде публикуван нов процес в облачното пространство с цел да бъде настроен по график, се натиска бутонът Publish в лявата горна секция на IBM RPA Studio.

Graphical user interface, application

Description automatically generated

Допълнение: при публикуване на нов процес в облачното пространство, следва да бъдат конфигурирани и така наречените Execution Times. Tе са отговорни за времетраенето на изпълнение на самият процес. Ако бъдат оставени с началните им стойности, това означава, че процесът ще се изпълнява в продължение на 2 минути, след което компилаторът автоматично ще прекъсне изпълнението и може да причини системна грешка, ако процесът е прекъснат преди неговият край. Тези времена могат да бъдат настроени още при самото публикуване на нов процес от IBM RPA Studio или след това от облачното пространство.

Първи метод (от студиото):

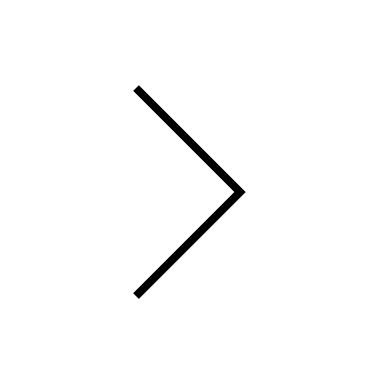
Graphical user interface, text, application

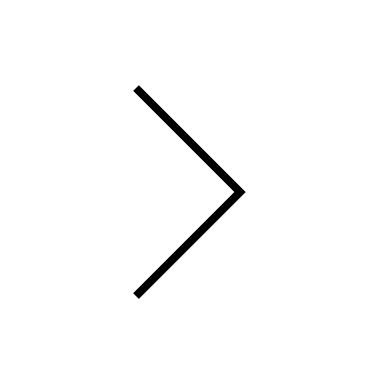
Description automatically generated

Втори метод (от облачното пространство):

В лявата колона избираме Scripts, след това избираме трите точки вдясно от името на скрипта и натискаме Details. В таб Versions отново ще видим името на нашият скрипт. Избираме трите точки вдясно от него и след това Edit execution times, откъдето следва да конфигурираме времената за изпълнение.

Graphical user interface, application

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

# Поддръжка на робота

## Поддръжка на дисковото пространство

Достатъчно дисково пространство трябва да се осигури за нормална работа на операционната система.

## Поддръжка на пощенската кутия

Роботът праща мейли през Outlook. Използвания Outlook профил трябва да се поддържа, за да предостави нормален достъп за изпращане на мейли.

# Преместване на робота на друга машина

## Операционни изисквания

### Хардуер и Софтуер

### https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct?deliverableId=A3386110AF3511EAB9117899910938C8

### Мрежови

### Достъп на Outlook до мейл сървъра за изпращане на мейли;

### Достъп до споделени файлове GoogleSheets;

* Достъп до приложението Azure;
* Достъп до приложението Jira.

## Инсталиране на IBM RPA Studio

### Инсталация

Влезте на https://eu1app.wdgautomation.com/ и отидете в меню About --->DownloadInstaller

Сваля се .exe файл, стартира се и се проследяват стъпките за инсталация (препоръчва се пълната инсталация на продукта)

### Активация

Същия лицензен ключ използван на старата машина може да се ползва за новата инсталация, но след като бъде освободен от старата машина. Като се посети <https://eu1app.wdgautomation.com/> ---> Configuration ----> Computers --->Edit Computer

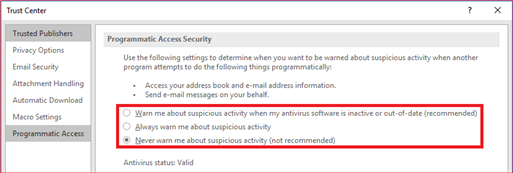
## Инсталиране и конфигуриране на приложенията

Microsoft Excel 2010 или по-нов, Outlook 2013 или по-нов и Internet Explorer 11 са необходими на новата машина за съвместимост с робота.

Конфигурирайте Outlook профила съответстващ на конфигурационния параметър „EmailCredentials“.

Активирайте програмния достъп до Outlook:

1. В Outlook кликнете File -> Options.
2. Кликнете Trust Center -> Trust Center Settings.
3. Кликнете Programmatic Access.
4. Изберете опцията „Never warn me about suspicious activity (not recommended)“.



Проверете успешното логване с акаунта на робота в приложенията Банкер и ТАН.

## Преместване на работните файлове

От старата машина или от архива копирайте директория „**[DataPath]**“ с всички директории и файлове на новата.

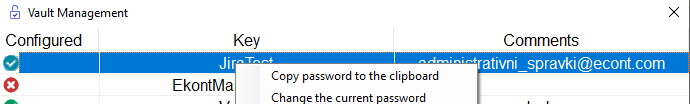
Конфигурационният файл трябва да бъде копиран в „*...\IBM RobotData*“.

Ако трябва работната директория да се сменя, се обърнете към разработчик, за да промени входните параметри за стартиране на процеса.

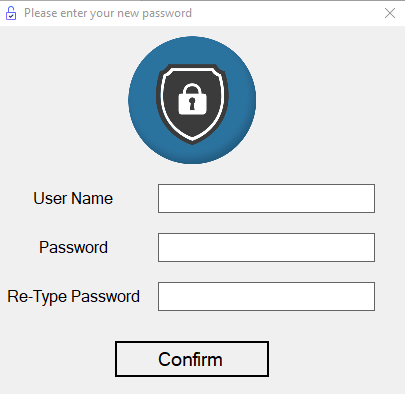
Конфигурационният файл трябва да се промени според новите изисквания и локация на директорията.

## Добавяне на Vault credentials

Влезте в <https://eu1app.wdgautomation.com/>

* ако са необходими нови Credentials ---> Configuration -----> New Vault Credential, след което се отваря Vault от Notification area на Windows Explorer , чрез десен клик на мишката и избор на меню Open Vault  (ако Vault е със статус Open се избира меню Manage ). В отворения прозорец Vault Manager за съответния Key се избира десен бутон на мишката и меню Change current password 

В новопоявилият се прозорец се вписват коректните име на потребител и парола и се натиска бутона Confirm за потвърждение на промените



* ако дадените credentials вече съществуват и са коректни при смяна на машината и свързване с акаунта в облачното пространство Vault Manager-а автоматично открива създадените потребителски данни.
  1. **. Добавяне на System Credentials**

Разликата между Vault Credentials и System Credentials се състои в това, че първите са конфигурирани локално на машината и в последствие биват извличани посредством Vault Manager-а в хода на изпълнение. System Credentials използват уеб базираният облак на IBM, като място за съхранение и извличане в последствие и за тази цел те изискват по специфичен метод за настройка.

Предварителни изисквания за конфигуриране:

* OpenSSL
* Персонален акаунт в уеб портала на IBM Cloud Platform
* Ръководство “Creating RSA Keys for Cloud Based Password Vault for WDG”

За да може роботът да извлича потребителските данни от облака, той се нуждае от така наречените RSA ключове, които се разделят на публичен и персонален, като създаването и конфигурирането им в облачното пространство е подробно описано в ръководството на IBM започващо от [страница 16 в този документ](#PublicAndPrivateKey).

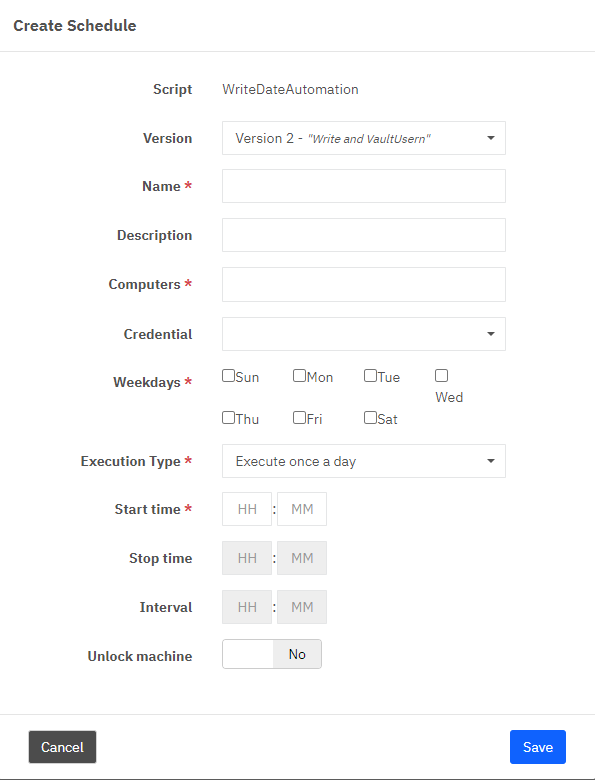
A picture containing parking, clock, meter, street

Description automatically generated

## 9.7 Задаване на график за стартиране на процеса

* Стартирането на процеса по график се конфигурира в <https://eu1app.wdgautomation.com/> ----> Scripts ---->  за съответният процес

---> 



Описание на полетата:

**Name:** тук именуваме графикът (задължително поле)

**Description:** кратко ориентировъчно описание

**Computers:** избираме името на машината, на която в последствие ще се запуска автоматизацията. Има възможност за добавяне на повече от една машина, като в последствие процесът ще се стартира едновременно на всички избрани в указаното време (задължително поле)

**Credential:** тук се избират потребителските данни за конкретната машина, като това поле се конфигурира единствено ако очакваме самата тя да ги използва, за да се логне автономно в последствие

**Weekdays:** избираме дни от седмицата, в които ще се стартира процесът

**Execution Type:** Execute once a day (ако искаме процесът да се стартира само веднъж на ден) Execute more than once a day (ако искаме процесът да се стартира повече от веднъж на ден)

**Start time:** тук задаваме стартово време

**Stop time:** това поле става активно, когато е избрана опцията Execute more than once a day. В указаният час приключва работата на робота за конкретният ден.

**Interval:** на база часове или минути се конфигурира през колко време да се стартира процесът

**Unlock machine:** в случай че машината е заключена при активиране на тази опция, тя сама се отключва при стартиран процес със съответните Credentials от Credential полето

Icon

Description automatically generated with low confidence

Creating RSA Keys for Cloud Based Password Vault for WDG

*Version 0.0*

*Rough Draft*

*Authors:*

*Ryan Sparks –* [*rmsparks@us.ibm.com*](mailto:rmsparks@us.ibm.com)

*Zach Silverstein -* [*zachary.silverstein@ibm.com*](mailto:zachary.silverstein@ibm.com)

# Introduction

This document will detail the steps to create a public and private key using OpenSSL creating a new credential in the web portal, and retrieving the newly created credential using WDG Automation Studio.

A picture containing parking, clock, meter, street

Description automatically generated

Figure 1. WBG Automation – Automation Studio

## Use case

Like many automation solutions you will need to access and coordinate with other applications and tools. Often times these tools will require some kind of credential to be used whether that be usernames, passwords, api keys, and any other sensitive data. It is best practice to avoid placing these credentials in scripts or places that can be exposed to unintended audiences. Using the cloud based vault that comes with WDG we can securely store and access these credentials through scripting to be used during automation.

## Prerequisites

* Personal WDG Automation Account activated
* Access to WDG Automation Web Portal
* Access to WDG Automation Studio
* Admin access in WDG Automation Web Portal
* OpenSSL installed on your computer

# Create the Public and Private Keys

Run OpenSSL, navigate to the target folder in which you want to store the key, and type the following command to create the private key. The name of the key is specified by the -out parameter.

*openssl genrsa -out privkey.pem 4096*

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

You will notice that a private key has been created and placed in the folder path specified by the command.

Next we will create the public key using the private key as input. Run the following command to create the public key. The name of the key is specified by the -out parameter.

*openssl rsa -in privkey.pem -out rsa.public -pubout -outform PEM*

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

For WDG to use the keys they need to be the correct file type. WDG supports the following encoding types for RSA keys.

Public: .pem, .xml, .der, .crt, and .cer

Private: Certificate, PEM, PKCS#12, and XML

We need to change the file type of the public key to a valid filetype. We will change it .pem as shown below.

A screenshot of a cell phone

Description automatically generated

# Upload Keys to WDG Automation Web Portal

Navigate to the WDG web portal and sign in using your credentials. Navigate to the Tenant Configuration by selecting the dropdown in the top left corner.

A screenshot of a cell phone

Description automatically generated

Select the Credential tab.

A screenshot of a cell phone

Description automatically generated

Upload the public and private keys that were created in the previous step. You will need to specify the directory location of the private key that exists on the host machine.

\*\*The public key **IS** shared by the Tenant and only needs to be uploaded a single time.

\*\*The private key is **NOT** is not shared across computers in the same tenant and will need to exist on every machine that intends to run bots that will access the cloud based password vault.

A screenshot of a cell phone

Description automatically generated

# Create Credential in WDG Web Portal

Create a credential in the WDG web portal. Navigate to the Credential section on the left had side.

A picture containing holding, bird, phone, player

Description automatically generated

Select “New Credential”

You will now enter the information for the credential you wish to create

**Name**: Name used to identify the credential.

**Username**: Username portion of the credential.

**Password**: Password portion of the credential.

A screenshot of a cell phone

Description automatically generated

After selecting “Save” it will create the new credential. We can now see it as an entry in the credential list.

A screenshot of a cell phone

Description automatically generated

# Access Credential from WDG Studio

We will create a simple script that will access and retrieve a credential from the cloud based password vault that was created in the previous step.

Open WDG Automation Studio and create a new .wal file.

A screenshot of a cell phone

Description automatically generated

On the left had side select the “Get Vault Item” action. We will update the parameters to pull the recently created credential.

Name: The name of the credential to be retrieved. In our case it is “Test Credential.”

System: This determines which vault to access. When toggled to true it will use the cloud based vault.

Success: Output variable used to determine retrieval success.

User Name: Username portion of the credential.

Password: Password portion of the credential.

A screenshot of a cell phone

Description automatically generated

We will now add logging to view the success and username variable output from the retrieval.

On the left had side select the “Log Message” action. Place the following in the message field

“Success: ${sucess}

UserName: ${userName}”

Because the password is a secure string we will not be able to log and view its value. It can be stored in a variable and used later in the script when the credential is needed.

A screenshot of a cell phone

Description automatically generated

Select “Save” and run the script.

A screenshot of a cell phone

Description automatically generated

Confirm that the logs show the credential retrieval is successful.

A screenshot of a cell phone

Description automatically generated