Robotic Process Automation

Development Specifications Document (DSD)

*Process Name: Calculate Client Security Hash*

Table of Contents

[**Document Overview**](#_fmc2ik42b62t) **1**

[Automated Master Project Details](#_soa72miybokv) **1**

[Runtime Guide](#_e5eh7vtp3elw) **1**

[Runtime Diagram](#_2pt89uzbsm6q) 1

[List of Packages](#_8uc76jjm25ud) 2

[Master Project Runtime Details](#_33q1drg667x0) 2

[**Project Details**](#_q7qb3l8qz84p) **3**

[Project Name: <project name>](#_vrc3lxjwb5na) 3

[Workflow(s) specific to the Project](#_in5ehl2op8tm) 4

[Project Name: <project name>](#_eddmfv9qxvte) 4

[Workflow(s) specific to the Project](#_cebp8lzhrwzt) 5

[**Compliance Considerations and Reporting Requirements**](#_azdzmbnsrqr4) **5**

[**Other Details**](#_gmvdjkbe065o) **6**

[Future Improvements](#_3e7irmfl1h6l) 6

[Debugging Tips](#_qtg3tsjmu03s) 6

[Other Remarks](#_zgfonke2bma) 6

[**Post UAT Specifications**](#_qba241jo7cu2) **6**

[**Glossary**](#_go2cr78yd0pl) **6**

Version Control

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Version | Role | Name | Organization Department | Function | Comments |
| 27-Dec-2020 | 1.0 | Author | Jes Hunsballe | N/A | N/A | Non-REF |
| 28-Dec-2020 | 2.0 | Author | Jes Hunsballe | N/A | N/A | REF version |



# Document Overview

The Development Specifications Document (DSD) is created for every business process automated using RPA. The DSD needs to be reviewed and updated for every change requested and applied to the automated process. This document provides a technical snapshot and must always reflect the latest design and key features of the automated workflow.

The document naming convention will follow the naming convention and the version of the automated process. This can be “business process name version” or it can be defined, case by case, as part of the larger RPA project design.

This document is completed by the RPA Solution architect and RPA developer who automates the business process. It is reviewed by the business process owner, application owner, and CoE design authority.

This document is meant to assist the RPA COE, IT operations and process owners by providing a snapshot of the automated process details and components. It can also serve developers to have a quick glance at the setup, before diving into the code, to troubleshoot or update changes. The purpose of the document is to record the outcome specific to the automated master project and its subcomponents: projects, workflows, sequences etc.

# Master Project Details

Details filled in by the developer reflect the actual information for the master project released for production.

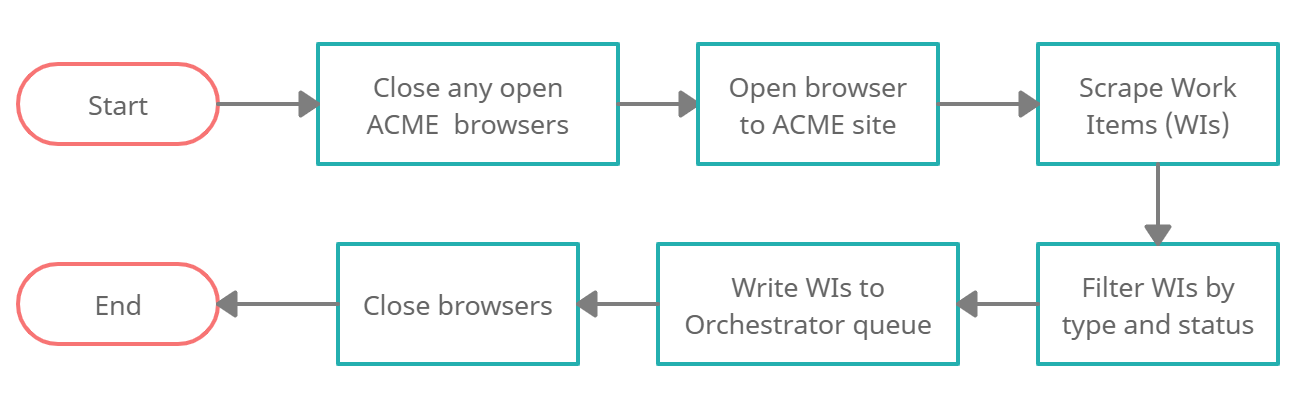
|  |  |  |
| --- | --- | --- |
| # | Item | Details  Fill in with free text. If not applicable, mark the filed as "N/A". No empty fields. |
| 1 | Master Project Name and Version | Calculate Client Security Hash v2 |
| 2 | Robot Type (attended/unattended/mix) | Attended |
| 3 | Is Orchestrator used? (Yes/No) | Yes |
| 4 | Scalable? (Yes/No)  Can the process be run by multiple robots in parallel? | Yes |

# Runtime Guide

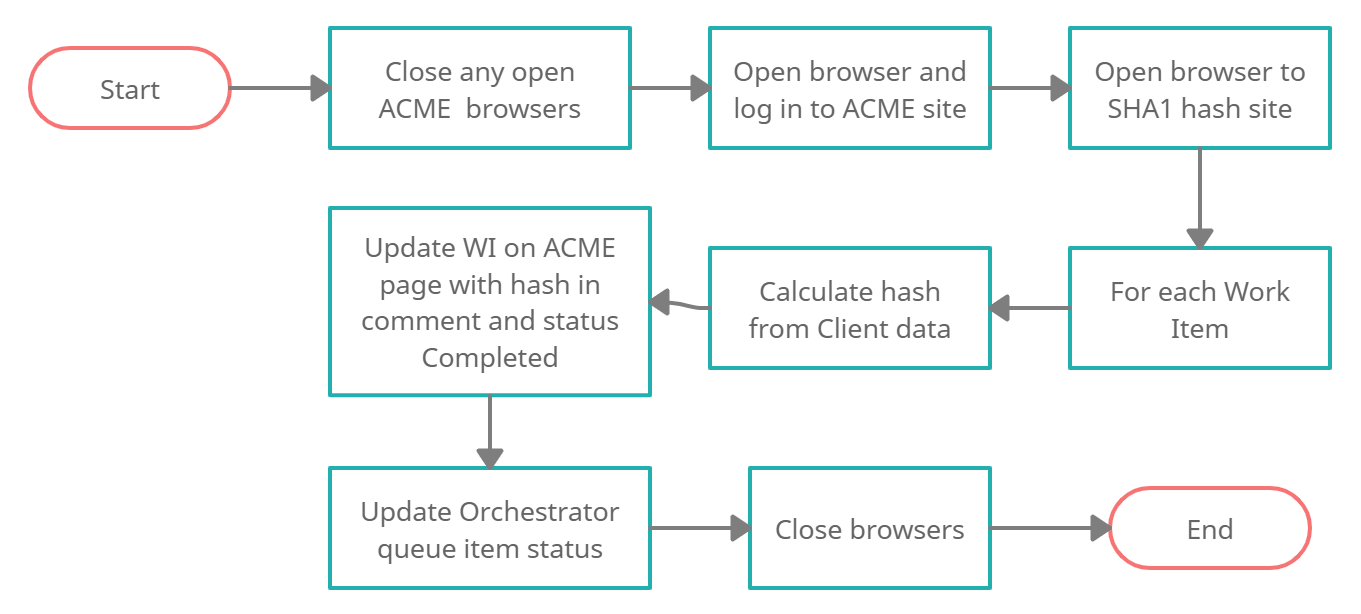
## Runtime Diagram

**Architectural Structure of the Master Project** Display the interaction between components (package / robots, Orchestrator queues, and running order).

**Dispatcher:**



**Performer:**

****

## List of Packages

Include **the list of packages and the high level description** for each of them, to explain each one's purpose:

|  |  |  |
| --- | --- | --- |
| # | Package Name | High-Level Description |
| 1 | UiPath.Excel.Activities v2.9.3 | N/A |
| 2 | UiPath.System.Activities v20.10.1 | System activities |
| 3 | UiPath.UiAutomation.Activities v20.10.8 | Browser activities |

\*Add more rows to the table to include all the project names and versions. No fields should be left empty. Use “N/A” for the items that don't apply to your project.

## Master Project Runtime Details

Details of the automated process:

|  |  |  |
| --- | --- | --- |
| # | Item | Details  (Fill in with free text. If the section does not apply to your automation, mark the field as “N/A”. No empty fields. ) |
| 1 | Production Environment Details | MS Windows, Chrome browser |
| 2 | Prerequisites to run | N/A |
| 3 | Input Data | Client ID, name, country (Work Item details) |
| 4 | Expected Output (output data) | Client security hash |
| 5 | How to start the automated process? | Manually by user from user’s Windows PC |
| 6 | Resuming the process from a particular step | Process is split in a Dispatcher and Performer. If either fail they will shut down and can be run from start again. |
| 7 | Reporting  queues reporting, Kibana or another platform | Queue items are updated when they are processed. |
| 8 | Manual Error Handling  roll back or manually complete failed transactions. Procedures to reset the item. Ex “set status as investigating” | If the dispatcher fails, it either uploads all or no queue items and the queue can be checked for a status.  If errors occur in the main process it can be run again. |
| 1. How to resume the process in case of error | Run the main process again and the failed transactions (Work Items) will be identified and processed. |
| 1. How to manually fix transactions with error | N/A |
| 9 | Use of Orchestrator |  |
| 1. Password Policies   specific compliance requests? | N/A |
| 1. Stored Credentials   Never hard code credentials in the workflow | Credentials to ACME test site are stored in Config.xlsx in Data folder. |
| 1. List of Asset Names | N/A |
| 1. List of Queues Name | WorkItemQueue |
| 1. Schedule Details | N/A |
| 10 | Recommended Resolution | N/A |

# Project Details

In this section describe all the projects that compose the automated process.

For each project, describe the workflow(s) in the logical order that they are called in.

If the workflow is a flowchart, also include the exported image from Studio.

If the automated process is composed of multiple projects, copy paste and fill in the table below for each project with its specific details (there are 2 here already, assuming a dispatcher and performer project)

## Project Name: Calculate Client Security Hash

General information about the process selected for RPA prior to automation.

|  |  |  |
| --- | --- | --- |
| # | Item Name | Details  Fill in with free text. If not applicable, mark the field as “N/A". No empty fields. |
| 1 | Environment used for development  name, location, configuration details etc | UiPath 2019.10.5 |
| 2 | Environment prerequisites  OS details, libraries, required apps | Windows 10, Chrome browser |
| 3 | Logging level | Trace logging with the RE Framework |
| 4 | Details about automation  if the apps were automated using UI Automation, Image & Text | UI automation via Chrome |
| 5 | In case of attended bot, can the user operate the computer while the robot is running? | No |
| 6 | Repository for project  where the developed project is stored | N/A |
| 7 | List of reused components | N/A |
| 8 | Custom logs defined in the workflows  where Throw Activity was used or custom log message was defined | Business rule exception Error level at ACME Login (incorrect email/password or page didn’t load).  Business rule exception Warn level if no work items of filtered type and status exist.  Any system exceptions in try/catch running “ACME/ACME\_ProcessWorkItem.xaml". |
| 9 | Frequent errors found in the development phase | N/A |
| 10 | Workarounds used in the automation phase | RE Framework used for Dispatcher and a dummy QueueItem is used (in Dispatcher\_Framework\GetTransactionData.xaml) to bypass the requirement of changing TransactionItem data type throughout the whole process as it only runs once. |
| 11 | Configuration method  assets, excel file, Json file | Excel file |
| 12 | Configuration details  path for input files, configuration Orchestrator assets used | Config file in Data/Config.xlsx |

### Workflow(s) specific to the Project

Define below all the workflow files (.xaml files) used in the project, with the Input and Output data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Workflow File Name | Description | Arguments | Comments |
|  | Dispatcher.xaml | Dispatcher workflow | in\_OrchestratorQueueName - string | RE Framework |
|  | Dispatcher\_Framework/CloseAllApplications.xaml | Ends processes, logs out and closes used applications. | In\_Config – dictionary | RE Framework |
|  | Dispatcher\_Framework/GetTransactionData.xaml | Gets a transaction item (all Work Items) from ACME test site. | In\_TransactionNumber – integer,  In\_Config – dictionary,  Out\_TransactionItem – queueItem,  Out\_TransactionField1 – string,  Out\_TransactionField2 – string,  Out\_TransactionFieldID – string,  io\_TransactionData - string | RE Framework |
|  | Dispatcher\_Framework/InitAllApplications.xaml | Open applications used in the process and do necessary initialization procedures (e.g., login). | In\_Config – dictionary | RE Framework |
|  | Dispatcher\_Framework/InitAllSettings.xaml | Initialize, populate and output config Dictionary to be used throughout the project where settings and constants are read. | In\_ConfigFile – string,  In\_Sheets – string[],  In\_Config – dictionary | RE Framework |
|  | Dispatcher\_Framework/KillAllProcesses.xaml | Force termination of the Windows applications used in the business process. | In\_Config – dictionary | RE Framework |
|  | Dispatcher\_Framework/RetryCurrentTransaction.xaml | Manage the retrying mechanism when a system exception occurs. | In\_Config – dictionary,  Io\_RetryNumber - integer,  Io\_TransactionNumber – integer,  In\_SystemException - exception,  inQueueRetry - boolean | RE Framework |
|  | Dispatcher\_Framework/SetTransactionStatus.xaml | Set and log the transaction's status and log fields. There can be three possible statuses: Success, Business Exception and System Exception. | In\_Config – dictionary,  in\_TransactionItem – queueItem,  Io\_RetryNumber - integer,  Io\_TransactionNumber – integer,  in\_TransactionField1 – string,  in\_TransactionField2 – string,  in\_TransactionFieldID – string,  In\_SystemException - exception,  In\_BusinessException - exception | RE Framework |
|  | Dispatcher\_Framework/TakeScreenshot.xaml | Capture a screenshot, log its name and location and save it with the PNG extension. | In\_Folder,  Io\_FilePath | RE Framework |
|  | Main.xaml | Dispatcher workflow | in\_OrchestratorQueueName - string | RE Framework |
|  | Main\_Framework/CloseAllApplications.xaml | Ends processes, logs out and closes used applications. | In\_Config – dictionary | RE Framework |
|  | Main\_Framework/GetTransactionData.xaml | Gets a transaction item (all Work Items) from ACME test site. | In\_TransactionNumber – integer,  In\_Config – dictionary,  Out\_TransactionItem – queueItem,  Out\_TransactionField1 – string,  Out\_TransactionField2 – string,  Out\_TransactionFieldID – string,  io\_TransactionData - string | RE Framework |
|  | Main\_Framework/InitAllApplications.xaml | Open applications used in the process and do necessary initialization procedures (e.g., login). | In\_Config – dictionary | RE Framework |
|  | Main\_Framework/InitAllSettings.xaml | Initialize, populate and output config Dictionary to be used throughout the project where settings and constants are read. | In\_ConfigFile – string,  In\_Sheets – string[],  In\_Config – dictionary | RE Framework |
|  | Main\_Framework/KillAllProcesses.xaml | Force termination of the Windows applications used in the business process. | In\_Config – dictionary | RE Framework |
|  | Main\_Framework/RetryCurrentTransaction.xaml | Manage the retrying mechanism when a system exception occurs. | In\_Config – dictionary,  Io\_RetryNumber - integer,  Io\_TransactionNumber – integer,  In\_SystemException - exception,  inQueueRetry - boolean | RE Framework |
|  | Main\_Framework/SetTransactionStatus.xaml | Set and log the transaction's status and log fields. There can be three possible statuses: Success, Business Exception and System Exception. | In\_Config – dictionary,  in\_TransactionItem – queueItem,  Io\_RetryNumber - integer,  Io\_TransactionNumber – integer,  in\_TransactionField1 – string,  in\_TransactionField2 – string,  in\_TransactionFieldID – string,  In\_SystemException - exception,  In\_BusinessException - exception | RE Framework |
|  | Main\_Framework/TakeScreenshot.xaml | Capture a screenshot, log its name and location and save it with the PNG extension. | In\_Folder,  Io\_FilePath | RE Framework |
|  | ACME/ACME\_LogIn.xaml | Logs in to ACME test site | In\_username - string,  In\_password - string | Sequence |
|  | Browser/OpenBrowser.xaml | Open Chrome browser at given url | In\_Url - string | Sequence |
|  | ACME/ACME\_IterateWorkItems.xaml | Extract Work Items, filter them and upload them to Orchestrator queue. | In\_URLAcmeWI – string,  In\_WIType – string,  In\_WIStatus – string,  In\_Config – dictionary,  In\_Transactionitem – queueItem | Sequence |
|  | ACME/ACME\_ProcessWorkItem.xaml | Process one queueItem (Work Item) and scraping Client Data, calculate hash and update the Work Item with hash and new status. | in\_URL\_AcmeWI – string,  in\_WIID – string | Sequence |
|  | ACME/ACME\_LogOutAndClose.xaml | Log out of and close any open ACME browser tabs. | in\_ACMEUrl - string | Sequence |

# Compliance Considerations and Reporting Requirements

# Other Details

## Future Improvements

* Set Dispatcher and Performer to run in Orchestrator with a daily trigger
* More robust clean up of left over SHA1 pages

## Debugging Tips

## Other Remarks

# Post UAT Specifications

* Average duration per transaction (varies depending on the Test environment):
* Recommended number of robots for the specified volumes:
* Specified schedule:

# Glossary

* **Master project** - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation.
* **Project** - a UiPath Studio project containing one or multiple workflow files. A project can be converted to a package and run independently, covering a particular scope within the master project. The project is used when defining the development and support phase of the automation.
* **Package** - the output of compiling a project. A package can be deployed on the robot machine and be executed by the robot service. Only one package can be executed at a given time by a robot. The package is used when defining the running phase of the automation
* **Workflow** - a component of the package, the workflow encapsulates a part of the project logic. The workflow can be of type: sequence, flowchart or state machine. a workflow is saved as an .xaml file inside the project folder. A workflow file can be invoked from another workflow and by default there is an initial workflow file that will run when executing the package.
* **Activity** - an action that the robot executes.
* **Sequence** - a workflow where activities are executed one after another, in a sequential order
* **Flowchart** - a workflow where activities are connected by arrows and the logic of the workflow can be easily followed in a visual manner. The flowchart can also be exported as an image from UiPath studio
* **State machine** - a more advanced way of organizing a workflow, similar to a flowchart.
* **BOR** - Back office robot
* **FOR** – Front office robot
* **Orchestrator** – Enterprise architecture server platform supporting: release management, centralized logging, reporting, auditing and monitoring tools, remote control, centralized scheduling, queue/robot workload management, assets management.