Development Specifications Document (DSD)

*Process Name: \*\*\**

Version Control

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Version | Role | Name | Organization Department | Function | Comments |
|  |  |  |  |  |  |  |



# 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.

# Automated 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 |  |
| 2 | Robot Type (attended/unattended/mix) |  |
| 3 | Is Orchestrator used? (Yes/No) |  |
| 4 | Scalable? (Yes/No)  Can the process be run by multiple robots in parallel? |  |

# Runtime Guide

## Runtime Diagram

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

## 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 [2.10.4] | Execute Excel related operations using the Open Office XML format (XLSX) or Excel.Interop |
| 2 | UiPath.System.Activities [21.4.1] | Core activities which enable the robots to manipulate data tables and collections, work with files and folders. Package also contains workflow operators, dialog forms, debugging and invoking methods. |
| 3 | UiPath.UIAutomation.Activities [21.4.4] | Package contains core activities which enable the automation of desktop applications, browsers, and virtual machines. Please check the documentation for more details. |
| 4 | UiPath.Python.Activities [1.1.6863.33404] | UiPath Python Activities Pack |
| 5 | UiPath.Testing.Activities [1.3.3] | Testing activities which enable users to verify expression values and control states in test case workflows. |

\*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 |  |
| 2 | Prerequisites to run |  |
| 3 | Input Data |  |
| 4 | Expected Output (output data) |  |
| 5 | How to start the automated process? |  |
| 6 | Resuming the process from a particular step |  |
| 7 | Reporting  queues reporting, Kibana or another platform |  |
| 8 | Manual Error Handling  roll back or manually complete failed transactions. Procedures to reset the item. Ex “set status as investigating” |  |
| 1. How to resume the process in case of error |  |
| 1. How to manually fix transactions with error |  |
| 9 | Use of Orchestrator |  |
| 1. Password Policies   specific compliance requests? |  |
| 1. Stored Credentials   Never hard code credentials in the workflow |  |
| 1. List of Asset Names |  |
| 1. List of Queues Name |  |
| 1. Schedule Details |  |
| 10 | Recommended Resolution |  |

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 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |

# Compliance Considerations and Reporting Requirements

* NA

# Other Details

## Future Improvements

* NA

## Debugging Tips

* NA

## Other Remarks

* NA

# 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.