@<Project-Name>

Solution Design Document (SDD)

Table of Contents

[1 INTRODUCTION 3](#_Toc30514748)

[1.1 Purpose 3](#_Toc30514749)

[1.2 Objectives 3](#_Toc30514750)

[1.3 Key Contacts 3](#_Toc30514751)

[1.4 History 3](#_Toc30514752)

[2 AUTOMATION DETAILS 4](#_Toc30514753)

[3 RUNTIME GUIDE 5](#_Toc30514754)

[3.1 Architectural structure of the Master Project 5](#_Toc30514755)

[3.2 Master Project Runtime Details 5](#_Toc30514756)

[3.3 Project(s) workflows 6](#_Toc30514757)

[3.4 Packages 6](#_Toc30514758)

[3.5 Architectural structure of the Master Project 7](#_Toc30514759)

[4 OTHER DETAILS 8](#_Toc30514760)

[5 GLOSSARY 9](#_Toc30514761)

# INTRODUCTION

## Purpose

The document contains the major components of the Master Project (the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation) taking into account all the business restrictions (scheduling, peaks, future increases in volume etc.).

## Objectives

The focus of the Solution Architect will be on:

Robustness

Scalability

Efficiency

Replicability

Reusability of components

The information herein is targeted primarily at the developers that will initially implement the solution and subsequently at the support developers in case of change requests.

## Key Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Contact Details | Notes |
|  |  |  |  |
|  |  |  |  |

## History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Created by | Role | Organization | Notes |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# AUTOMATION DETAILS

Details filled in need to reflect the actual information for the Master Project released for production. The following table will be populated:

|  |  |
| --- | --- |
| Item | Description |
| Master Project Name | @<Project-Name> |
| Robot Type | e.g. BOR, FOR or Mix |
| Orchestrator used? | @<IsBackground> |
| Scalable | e.g. No |
| UiPath version used | @<studioVersion> |

# RUNTIME GUIDE

## Architectural structure of the Master Project

Display the interaction between components (package / robots, Orchestrator queues, and running order) in a diagram

## Master Project Runtime Details

Outlines the details of the automated process by filling in the table below.

|  |  |
| --- | --- |
| Item | DESCRIPTION  *Fill in each bolded section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.* |
| Production environment details | Example: Running on Sparky , the virtual backoffice machine. Scheduled every night after the report is generated from Zendesk. |
| Prerequisites to run | Example: Report was generated by Zendesk |
| Input Data | Email received in Zendesk\_reporting@uipath.com |
| Expected output | Having Excel on the machine |
| How to start the automated process | Example: 3 valid CSV files |
| Reporting | 2 source files in C:\ZendeskReporting |
| (queues reporting, Kibana or another platform) | Example: 2 e-mails sent to e-mail address: management@uipath.com |
| How is Orchestrator used? | Example: 3 valid CSV files |
| Password policies | 2 source files in C:\ZendeskReporting |
| (mention any specific compliance requests) | Example: 2 e-mails sent to e-mail address: management@uipath.com |
| Stored credentials | Example: The process will be started from orchestrator server (demo.uipath.com) |
| (Never use hardcore credentials in the workflow!) |  |
| List of queues names | Example: Orchestrator logs and jobs dashboards. |
| (Naming convention: ProcessName\_QueueName) | Example: Orchestrator used for scheduling and asset passwords. |
| Schedule Details | Example: G-mail password only, not expiring. |
| Multiple Resolutions Supported?  (in case of image automation / Citrix and VDI) | Example: Stored in Orchestrator Assets |
| Recommended Resolution | Example: n/a |

Project name 1

|  |  |
| --- | --- |
| Item | DESCRIPTION  *Fill in each bolded section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.* |
| Environment used for development  (name, location, configuration details etc) | Example: DEV\_Env1\_EMEA ( UiPath computer) |
| Environment prerequisites  (OS details, libraries, required apps) | Example: Windows 7, BackOffice&Studio license, Microsoft Excel |
| Repository for project  (where is the developed project stored) | Example: \\myshare.com\Zendesk |
| Configuration method  (assets, excel file, Json file) | Example: Assets |
| List of reused components | Example: found via MyGo! |
| List of new reusable components | Example: placeholders created in MyGo! |

Add tables for as many projects as you need and fill them in.

## Packages

Include the list of packages and high-level description for each of them, to explain their purpose

@<Dependency-Package>

## Architectural structure of the Master Project

Display the interaction between components (package / robots, Orchestrator queues, and running order) in a diagram.

# Project Workflows

@<WorkFlowFiles>

# OTHER DETAILS

Future Improvements

Fill in any improvements that need to be considered for the future:

Example:

Optimize the processing algorithm

Implement process error recovery (retry)

Enable support for multiple template files

Other Remarks

Please mention here any other points that you consider relevant for the automation process.

Example: The workflow should run every night at 7AM. Be careful not to schedule it before the report is generated by Zendesk.

The Zendesk generated data is always 1 day old.

# GLOSSARY

The main terms used in the Solution Architecture Document are defined below:

**Master project** - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation. There is a 1 to 1 connection between the Master Project and the Process to be automated (As presented in the PDD).

**Project** - an 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. Or multiple projects can be converted into one package depending on the aims and restrictions of the automation. The project is used when defining the development and support phase of the automation.

**Package** - the output of compiling one or multiple projects. 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.