



Single Bot Framework

Read Me

Table of Contents

1. Introduction	3
1.1 Overview.....	3
1.2 Common Use cases	3
2. Requirements & Prerequisites	4
2.1 System Requirements	4
2.2 Prerequisites	4
2.3 Security Measures	4
2.4 Disclaimers	4
3. Getting Started	5

3.1	Skill Matrix	5
3.2	Installation Hierarchy	5
3.3	Quick Start	5
3.3.1	Setup	5
3.3.2	Configuration	5
4.	Reports	9
5.	Logs	10
6.	Troubleshooting & Support	11
6.1	Support.....	11
6.2	FAQs	11
	Appendix A: Record of Changes.....	13
	Appendix B: Acronyms	14
	Appendix C: References	15

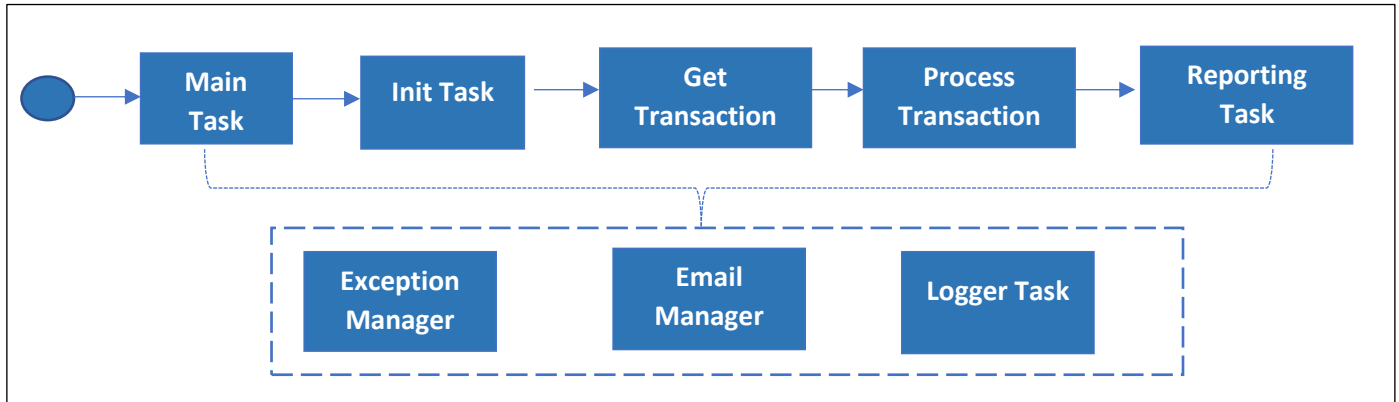
1. Introduction

This document contains all essential information for the user to make full use of the Single Bot Framework. This manual includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the Bot.

1.1 Overview

Description - It sets up a standard for Bot architecture for cases where the input to the Bot is received in bulk (excel/Database) and the bot process records sequentially

Diagrammatic representation of the flow of Framework Components.



- When a Bot is triggered the flow starts from Main Block (Parent Task). Every child task shown above is triggered from Main Task.
- Exception occurred in any child Task is thrown to Main task with exception reason, wherein the same is handled by triggering Exception Manager Task.
- Logger Task is called whenever any Log needs to be captured.
- Emails are triggered by Email Manager Task.
- Init Task has below Child Tasks –



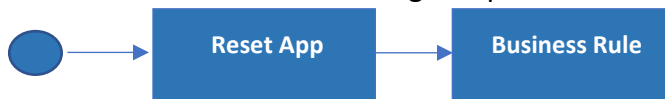
These child tasks are reusable and can be plugged into any project with minor adjustments.

- In Get Transaction block the logic to read Input Data is incorporated



These child tasks are used to read input source from related Input source.

- In Process Transaction Block the logic to process the Input data is incorporated.



Reset App Child Task is used to get the Applications to Home Page while processing second record onwards

Business Rules tasks will include any logic being applied on Input records.

- In Reporting Task, the logic to prepare the Summary Report is incorporated

Use Case - The framework can be used in any Automation Anywhere process with Single Bot architecture

2. Requirements & Prerequisites

2.1 System Requirements

- Not Applicable

2.2 Prerequisites

- Automation Anywhere Enterprise v11 or above.

2.3 Security Measures

- Not Applicable

2.4 Disclaimers

- Not Applicable

3. Getting Started

3.1 Skill Matrix

Instruction: This is a template for single Bot Architecture

3.2 Installation Hierarchy

<AA Directory>

- My Tasks
 - o Single Bot Framework - CitiusTech (Folder)
 - ✦ Error Folder
 - Logs (Folder)
 - o Error Logs Month-Day-Year.txt
 - Snapshots (Folder)
 - o Error Snap Month-Day-Year HoumMinuteSecond.png

- ✦ Input Folder
 - ProcessName_ConfigFile.xlsx

- ✦ Output Folder

- ✦ My Scripts

- ✦ My Tasks
 - 1.ProcessNameMain.atmx
 - 2.ProcessNameInit.atmx
 - 2.1. 0ProcessNameKillTask.atmx
 - 2.2.0_ProcessNameReadConfig.atmx
 - 2.3.0_ProcessNameEnvironmentCleanup.atmx
 - 2.3.1_ProcessNameCacheClear.atmx
 - 2.3.2_ProcessNameLogArchive.atmx
 - 2.4.0_ProcessNameInitialize.atmx
 - 3.0_ProcessNameGetTransaction.atmx
 - 3.1_ProcessNameReadFrom (Source Name). atmx
 - 3.2_ProcessNameUIScrapping.atmx
 - 4.0_ProcessNameProcessTransaction.atmx
 - 4.1_ProcessNameResetApp.atmx
 - 4.2_ProcessNameBusinessRules.atmx
 - 5.0_ProcessNameExceptionHandler.atmx
 - 6.0_ProcessNameLogger.atmx
 - 7.0_ProcessNameGenerateReport.atmx
 - 8.0_ProcessNameEmailManager.atmx

3.3 Quick Start

3.3.1 Setup

User would need to run the installation file (Single Bot Framework.msi)

3.3.2 Configuration

Input Variables/Arguments:

Config Path

Output Variables/Arguments:

IsSuccess (By Default): True/False

Exception Type: System/Business

3.3.3 Task Files

*** -- Denotes more description about the task below

Sr.No.	File Name	Description
1	1.ProcessNameMain	This is the Main Task
2	2.ProcessNameInit	This is the Parent Initialize task and child to Main
3	2.1.0ProcessNameKillTask	This is a child task to Init to kill all applications before starting the process ***
4	2.2.0_ProcessNameReadConfig	This is a child task to Init to read the Config File and put the data into an array ***
5	2.3.0_ProcessNameEnvironmentCleanup	This is a child task to Init to clean environment to avoid any failures
6	2.3.1_ProcessNameCacheClear	This is a sub Child Task to Environment Cleanup to Clear Cache of the Websites
7	2.3.2_ProcessNameLogArchive	This is a sub Child Task to Environment Cleanup to archive Log files for more than 30 days
8	2.4.0_ProcessNameInitialize	This is Child task to Initialize as in Launch websites or open any application involved
9	3.0_ProcessNameGetTransaction	This is the Parent task to Get data and child to Main
10	3.1_ProcessNameReadFrom (Source Name)	This is Child task to Get Transaction – to read data from the i/p Source and put the data into array
11	3.2_ProcessNameUIScrapping	This is Child task to Get Transaction – to read and extract data from UI if applicable
12	4.0_ProcessNameProcessTransaction	This is the Parent task to Process data and child to Main
13	4.1_ProcessNameResetApp	This is a Child task to Process Transaction to Reset Application (that is to bring any applications to Home page, shall be

		necessary in processing the consequent records)
14	4.2_ProcessNameBusinessRules	This is a Child task to Process Transaction. This block will include any logic for Processing the records
15	5.0_ProcessNameExceptionHandler	This is a child task to Main, and handles the exception task ***
16	6.0_ProcessNameLogger	This is a child task to Main, and handles the Logger Task ***
17	7.0_ProcessNameGenerateReport	This is a child task to Main, and will include the logic to create Report
18	8.0_ProcessNameEmailManager	This is a child task to Main, and will include conditions to send Email and accomplish the task

Task Details

ProcessNameKillTask - Command prompt function – taskkill.exe has been used to Kill Tasks. The list of exe’s to be killed is listed in the variable vKillTaskApps in the Variable Manager or can be listed in the Config File in the below shown example

e.g. – vKilltaskApps -> iexplore.exe /im excel.exe

ProcessNameReadConfig – The Config File is connected as a Database using a Connection String and then the data is stored in a Config Array.

Config Array is passed throughout the BOT and the array is looped to store the required value in a variable in the corresponding Child Tasks.

ProcessNameExceptionHandler – Exception Handling is done Globally. Every ATMX file is wrapped in an exception Block and at the end there is an IF Block where the Exception Line Number is checked for value more than ‘0’. If so, then the Exception Manager task is called with the exception details captured in variables and then the control flows to the next line in the parent task where there is an another IF Block to check if there has been any exception caused in the child task; if so then the Stop Task Command gets executed flowing to the corresponding Parent Task.

In the Exception Handler task the Error is logged in the Error log text file and snapshot in the snapshot Folder in Exception Log Folder

ProcessNameLogger – In the main task a variable is initialized with the current Date & Time. This variable is passed to the Logger task to create a text file with the captured Date & Time. Throughout

the BOT where ever Logging needs to be done the message is stored in a Variable through Variable Operation and the Logger task is called wherein the message gets logged in a predefined manner mention below

(Current Timestamp) | Project Name | Process Name | Process ID | *****Log Message*****

3.3.4 Folder Structure

Metabot Folder - \$AAApplicationPath\$\Automation Anywhere\My MetaBots**Project Name**

Scripts Folder - \$AAApplicationPath\$\Automation Anywhere\My Scripts**Project Name**

Tasks Folder - \$AAApplicationPath\$\Automation Anywhere\My Tasks**Project Name\Process Name**

Documents Folders - \$AAApplicationPath\$\Automation Anywhere\My Docs**Project Name\Process Name\Input**

\Output

\Reports

\Temp

Logs Folder - \$AAApplicationPath\$\Automation Anywhere\My Logs**Project Name\Process Name\Exception Log\Error Log**

\Snapshots

\Process Log

4. Reports

Not Applicable

5. Logs

- Error Logs Month-Day-Year.txt: In case of any error, this file logs error message along with time stamp
- Error *Snap Month*-Day-Year.png: In case of any error, this file captures screenshot of error

6. Troubleshooting & Support

6.1 Support

Not Applicable

Appendix A: Record of Changes

Instructions: Provide information on the version number, the date of the version, the author/owner of the version, and a brief description of the reason for creating the revised version.

No.	Version Number	Date of Change	Author	Notes
1	1.1	07/08/2020	Akriti Mahapatra	

Appendix B: Acronyms

*Instructions: Provide a list of acronyms and relative description used within the document.
Sequence should be alphabetical*

No.	Acronym	Description

Appendix C: References

No.	Topic	Reference Link

