Date: 20 May 2020

Celebrating 20 Years

Essentials Enquiry Bot – Readme

Version: 1.1
Date: 20th May 2020

A Larsen & Toubro
Group Company

Date: 20 May 2020

Bot Overview

The current COVID 19 outbreak has resulted into a nationwide lockdown in most of the countries, forcing people to stay indoors and work remotely, making it further difficult to procure even the basic essential items and causing them to constantly search for these supplies across different online stores or local shops that are usually running low or out of stock. LTI's Essential Enquiry Bot automates this process and diligently your employees to locate their essential items that are available at their preferred online stores within their zip codes. This bot not only helps them to stay home and stay safe but also saves a lot of their time that would've spent in long supermarket queues or frequent online searches, thus, increasing their productivity while they safely work from home. This bot performs the following:

- Employee submits a form with the ZIP Code, Preferred Stores (e.g. Walmart, Target, Amazon) and preferred essential items (Medicines, Paper towels, Surface wipes etc.) through web-forms (like the Microsoft Forms).
- The Essentials Enquiry Bot accesses these store websites and gets the stock data from the store for a specific zip code or the nearest one available in their area.
- The bot then sends a notification to the employee as per the predefined time with the list of essential items and the respective store addresses.
- The employee can accordingly check-in at their store of choice and procure the required essential items

Pre-Requisites

- Automation Anywhere Enterprise v11.x
- Google Chrome browser
- Setup the resolution of the bot runner VM to 1980*1020.
- The zoom of the Google Chrome browser MUST BE 100%
- Enable AA plugin in Chrome
- Set up all config paths in the database.
  - VerySmallDelay
  - SmallDelay
  - MediumDelay
  - LargeDelay
  - WebsiteDelay
  - vMyTaskFolderPath
  - vMyDocsFolderPath
  - vMyScriptsFolderPath
  - vEasternMailTime
  - vCentralMailTime
  - vMountainMailTime
  - vPacificMailTime
  - vInternalSubscribeFileName
  - vInternalUnsubscribeFileName
  - vExternalSubscribeFileName
  - vExternalUnsubscribeFileName
- Enable SMTP access for Bot-runner Machine
- Enable SMTP config for Automation Anywhere

- Set the image in 'My Documents > Pictures' path for the email
- Make sure database connectivity is enabled (SSMS installed) on VM.
- Download and Install Access DB 2010 32 bit

Enter the machine name in the config table in the database. (for each bot runner in the 'Bot Configuration Table' as provided in the 'EEB_DatabaseSchemaConfiguration.SQL')

- Install python on bot runner VM: pip – pyodbc
- Install ODBC drivers for MSSQL Server.
- Change DB credentials in the 'EEB_LoadBalancer.py'.
- Schedule time in Control room should be in sync with the database in 'Bot Configuration Table'.
- Enable the auto-login feature on all the bot runners.
- Scaling change: https://pureinfotech.com/fix-windows-10-dpi-scaling-virtual-machine-high-definition-displays/

Installation

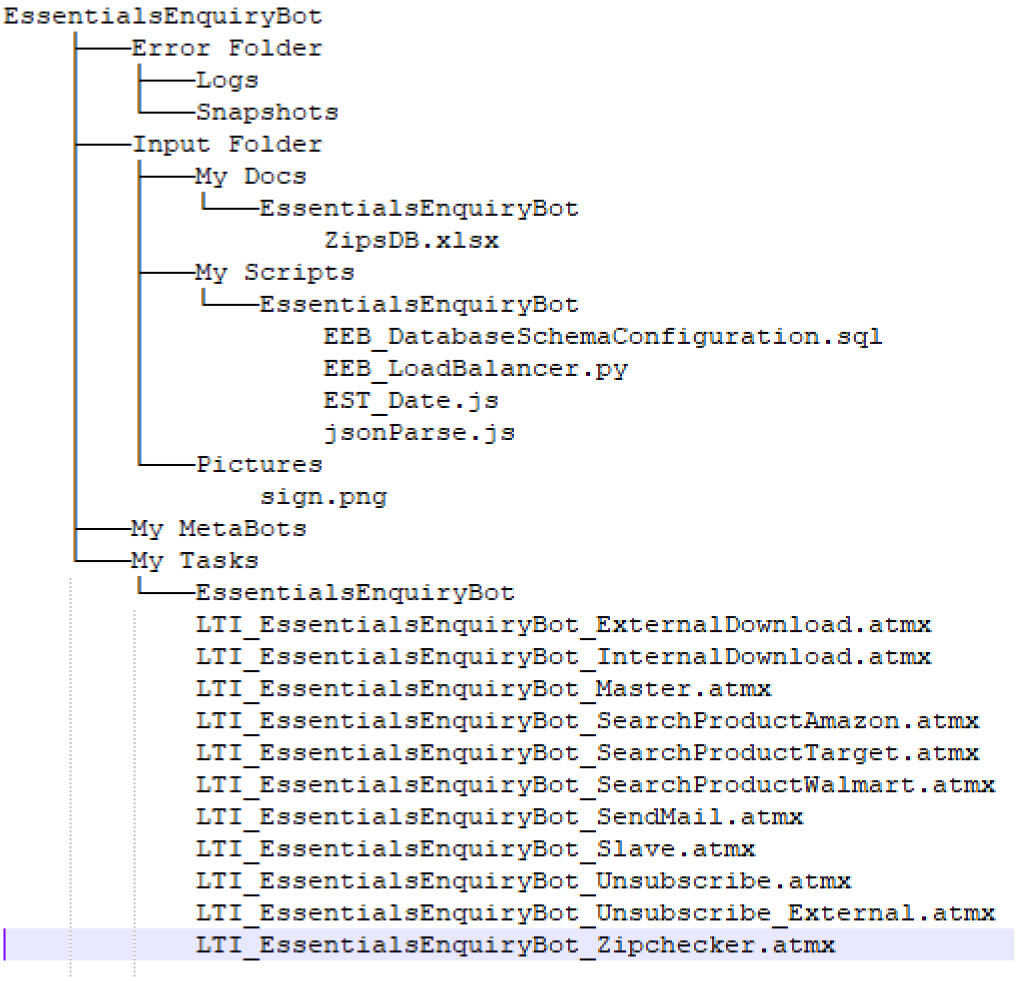
- Download the bot from Bot Store.
- Double click on the .msi file.
- On Welcome to the Installation wizard, click Next to continue.
- Click 'I agree to the terms in the license agreement' radio button to accept the agreement.
- Get/Copy the License key from Bot Store Downloads into License Key, click Next to continue.
- Click Install to begin the installation.
- Click Finish to complete the installation.
- To view the installation, go to 'My Tasks' folder on the AAE Client to see bot files.

Automation needs processing data in a standard template. This data can be generated using web-based forms or options like Microsoft forms. Data collection mechanism is user-dependent, but the input must be in the standardized format. Format used by the automation is shared in 'Internal_EEB_Subscription_Data.xslx' and 'External_EEB_Subscription_Data.xslx' files.

Uninstall

- Open Add/Remove Programs
- Select the Bot/Digital Worker to be installed.
- Click uninstall.

Create the following folder structure of the content in the AA Dir:



```
EssentialsEnquiryBot
    ├───Error Folder
    │   ├───Logs
    │   └───Snapshots
    ├───Input Folder
    │   ├───My Docs
    │   │   └───EssentialsEnquiryBot
    │   │           ZipsDB.xlsx
    │   ├───My Scripts
    │   │   └───EssentialsEnquiryBot
    │   │           EEB_DatabaseSchemaConfiguration.sql
    │   │           EEB_LoadBalancer.py
    │   │           EST_Date.js
    │   │           jsonParse.js
    │   └───Pictures
    │           sign.png
    ├───My MetaBots
    └───My Tasks
        └───EssentialsEnquiryBot
                LTI_EssentialsEnquiryBot_ExternalDownload.atmx
                LTI_EssentialsEnquiryBot_InternalDownload.atmx
                LTI_EssentialsEnquiryBot_Master.atmx
                LTI_EssentialsEnquiryBot_SearchProductAmazon.atmx
                LTI_EssentialsEnquiryBot_SearchProductTarget.atmx
                LTI_EssentialsEnquiryBot_SearchProductWalmart.atmx
                LTI_EssentialsEnquiryBot_SendMail.atmx
                LTI_EssentialsEnquiryBot_Slave.atmx
                LTI_EssentialsEnquiryBot_Unsubscribe.atmx
                LTI_EssentialsEnquiryBot_Unsubscribe_External.atmx
                LTI_EssentialsEnquiryBot_Zipchecker.atmx
```

How to Use the BOT?

This bot uses all the configurations from the Credential Manager 'ConnectionString' and the database. There are no IOs at the task level for this automation. Details about different parameters and database entries are provided in the pre-requisites section. Developer team of the automation can be contacted through 'IntelligentRPACoETeam@lntinfotech.onmicrosoft.com' email address.

Notes:

- This bot depends on the input data as per the standardized format. Operational data was in the excel generated by Microsoft Forms for a typical implementation.
- Email content and unsubscribe related details can be updated in each email body for any customization. (Currently this functionality is implemented, provide unsubscribe screenshot)
- Automation needs processing data in a standard template for unsubscribe functionality. This data can be generated like subscription using web-based forms or options like Microsoft forms. Data collection mechanism is user-dependent, but the input must be in the standardized format. Automation will pick up those details from file as per format in the 'Internal_EEB_Unsubscription_Data.xslx' and

'External_EEB_Unsubscription_Data.xslx' file. Users in this file will not get notification after next execution of the automation.

Bot Dependency Files Details:

- Automation maintains a copy of validated zip codes locally in 'ZipsDB.xslx' file for better performance and 'jsonParse.js'.
- Load balancing in different bot runners is managed by python script 'EEB_LoadBalancer.py'
- 'EST_Date.js' is used for time conversion to EST time-zone.
- 'sign.png' image is used in the email body.

Error Handling

- The automation captured all the exceptions and logged those exceptions. Bot follows the below-mentioned hierarchy to store details of the exceptions.
    - **Error Folder**
        - **Logs:** Logs of all the level are logged into a log file Log-<<RUN_ID>>. The RUN_ID is formed with a combination of Day, Month, Year, Hour, Minute and Millisecond of the start of execution of the bot. In case of any error, this file logs error message along with time stamp, line error number and the task name.
        - **Snapshots:** Error Snap file 'ErrorSnap-Month-Day-Year-Hour-Min-Seconds.png' is created for every exception in the automation. In case of any error, this file captures screenshot of error.
- Task Status of bot is set to failed in case of error.
- Every bot should be configured to send email on failure from the AA Control Room. This is to cove the failure in the bot initialization.