Perform Various Email Operations-Automation Anywhere-Readme

Bot Overview

This bot performs the following:

- Performs several Microsoft Outlook Operations. Description of all the operations can be found in this document.

Pre-Requisites

- Automation Anywhere Enterprise v11.3.x

Installation

- Download the bot from Bot Store.
- Double click on <Bot Name>.msi and follow the installation instructions.

  For first time users, the "Bot Store" folder is created under <AA Directory>/My Tasks (on your local disk).

- ***Installer creates the following folder structure with content under the <AA Directory>***

<AA Directory>
- My Tasks
  - Bot Store
    - Perform Various Email Operations-Automation Anywhere (Folder)
      - My Tasks
        - OutlookV2MBOT_ReadAppointmentSample.atmx
        - OutlookV2MBOT_ReadMailSample.atmx
        - OutlookV2MBOT_ReadMeetingRequestSample.atmx
        - OutlookV2MBOT_ReadMailAsPDFSample.atmx
        - OutlookV2MBOT_SendMailSample.atmx
      - Error Folder
        - Logs (Folder)
          - Error Logs Month-Day-Year.txt
        - Snapshots (Folder)
          - Error Snap Month-Day-Year.png
      - Input Files
      - Output Files
- My MetaBots
  - OutlookV2.mbot

How to Use the Bot:

Use the following information to configure your bot:

Notes

1. There is no need to open any SMTP, IMAP or POP3 ports for this Outlook Metabot to work, since it interacts with the Outlook Application. Ensure that the Outlook Application opens and works without any issues or prompts for input. In case of latter use Object Cloning to get through.

2. Ensure that you have at least one Outlook Mail Profile configured on the machine (Bot Runner/Bot Creator) where the Automation Task will run. Go to Control Panel → Mail → Show Profiles.
3. "LaunchOutlook" Function will take approximately 30 seconds to complete. During this time Outlook Application window will Minimize and then Maximize. You will notice inactivity during this window, still, DO NOT interrupt the process.

4. All the "Close****" Functions will take approximately 5 seconds to complete. You will notice inactivity during this window, still, DO NOT interrupt the process.

5. While passing "AccountName" value use exactly what you see in Outlook Application. Mostly it will be your Email Address. If you're not sure then use GetAccountInformation Function to get details of your configured Accounts.

6. Metabot Functions/Logics from which you do not expect to receive any Value will still return an

   Output string. If the operation was successful, then its value will be "Success". If the operation failed, then its value will be "[ERROR]:<error description>".
7. All Functions/Logics that accepts "NumberOfItems" parameter and retrieves a collection of Mails, Appointments or Meeting Requests, fetches newest items and not the oldest.
8. Convention for FolderPath:

Format:	\\<AccountName>\<FolderName>\<SubFolderName>
Examples:	\\John.Doe@AutomationAnywhere.Com\Projects\Archive
	\\John.Doe@AutomationAnywhere.Com\Sent Items

9. Before you can Get/Read Info or Properties of a Mail, Appointment or Meeting Request by calling corresponding methods, e.g. GetSenderSMTPAddress, GetAppointmentDurationInMinutes, GetMeetingRequestBody, etc., you will need to first call the SelectMailItem, SelectAppointment or SelectMeetingRequest,GetSmtpAddressURL one time to preselect a specific item you are working with. Once an item is selected then you can call any number of Get/Read Functions following that. Refer to the sample tasks for example. Credential vault needs to be configured for GetSmtpAddressURL parameters.
   Credential Name – SetSMTPAddress, Attribute name & value below
   The value of SetPR_SMTP_ADDRES-"http://schemas.microsoft.com/mapi/proptag/0x39FE001E"
   The value of SetPR_SENT_REPRESENTING_ENTRYID-
   "http://schemas.microsoft.com/mapi/proptag/0x00410102"
10. Functions that contain "Array" in their Name return a Single-Dimension Array, that is, 'N' number of Rows and 1 Column. Use Array Type Variable to map to Output Parameter of such Functions.

11. Some Functions/Logics may seem like Duplicates, but, do compare their Input-Output Parameters to see if there is any difference. Or those may be there to maintain backward compatibility with previous Version(s) of this Metabot.

12. DownloadAttachmentPath should be an accessible and valid fully qualified Folder Path that exists. Also, make sure that the Windows User running the Task has Read & Write privileges for that Folder.

13. Once a Meeting Request (received in Inbox) is Accepted/Tentative and saved on Calendar, it is referred to as "Appointment". After responding, Meeting Requests are moved to "Deleted Items" folder. All entries present on Calendar are referred to as "Appointment.

14. You will need to construct Filter value in a certain format to pass to "****ByFilter" Functions. You can read following article to learn more or search for more info on Google: https://msdn.microsoft.com/en-us/library/microsoft.office.interop.outlook._items.find.aspx

    c. Mail Example:

    [ReceivedTime] > '11/13/2017 12:00 AM' And [ReceivedTime] < '11/17/2017 12:00 AM'

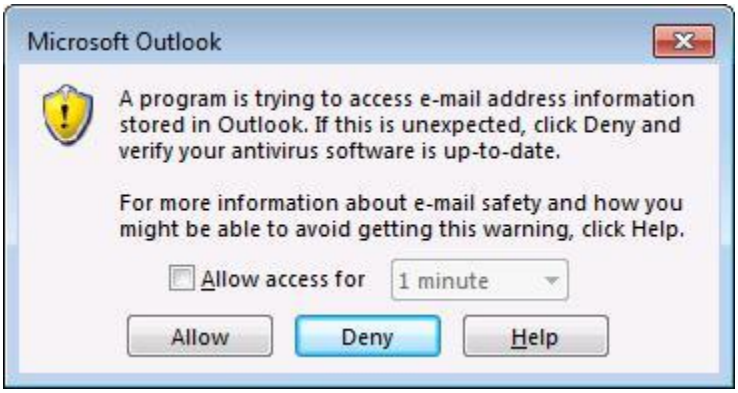
    b. Appointment Example:

    [Start] > '11/13/2017 12:00 AM' And [Start] < '11/17/2017 12:00 AM'

    c. Meeting Request Example:

    [ReceivedTime] > '11/13/2017 12:00 AM' And [ReceivedTime] < '11/17/2017 12:00 AM'

15. If you encounter an Error or a *Message Box* as shown below while using *SendMail* Function or while accessing Protected Properties, such as, Email Body, Email Sender Address, etc., then you will need to modify Outlook Settings using an Administrator User Account.



Outlook Application

Function	Inputs	Outputs	Comments
LaunchOutlook	N/A	String – Success	Launches Outlook Application. It is an optional step and only required if you need Outlook Application Visible. Else you could directly call SelectProfile Function.
SelectProfile	N/A	String – Success	Selects the current profile for the account that is logged into Outlook. Execute this Directly if you don't need Outlook Application open and visible. No Username or Password required to be configured for this.
SelectProfileByNamePassword	String – Profile Name, String – Profile Password	String – Success	Selects indicated Profile specifically. Create Password from Credential bot and pass on to metabot. Details provided below
SelectAccount	N/A	String – Success	Selects the default account for the selected profile
SelectAccountByName	String – Account name (user.name@email.addr)	String – Success	Selects the indicated account for the selected profile. Same as SelectAccount(AccountName)
SelectInbox	N/A	String – Success	Selects the Inbox folder for selected account
SelectCalendar	N/A	String – Success	Selects the Calendar folder for selected account
SelectFolder	String – FolderPath (\\user.name@email.addr\Folder\SubFolder)	String – Success	Selects the Folder indicated.

Try the two Resolution Methods described in following articlehttps://support.microsoft.com/en-us/help/3189806/a-program-is-trying-to-send-an-e-mail-message-on-your-behalf-warning-

GetAccountInformation	N/A	String – Account Information	Returns: DisplayName, UserName, SmtpAddress, and AccountType for all the Accounts under selected Profile.
GetFoldersArray	N/A	Array – Folders	Returns Array containing all Folders for selected Account.
CloseOutlook	N/A	String – Success	Closes all Outlook Application windows.
GetSmtpAddressURL	String – URL. Credential Name – SetSMTPAddress The value of SetPR_SMTP_ADDRES- "**http://schemas.microsoft.com/mapi/proptag/0x39FE001E**" The value of SetPR_SENT_REPRESENTING_ENTRYID - **"http://schemas.microsoft.com/mapi/proptag/0x00410102"**	N/A	Required for using all the get functionalities as discussed above at point 9. To be setup at Credential vault end. Guidelines below

GetSharedFoldersArray	N/A	Array – Shared Folders	Returns Array containing all Shared Folders the User has access to and are added to Outlook Profile. Use GetFoldersArray Function for folders in your Primary/Default Email Account.

Mails

Function	Inputs	Outputs	Comments
SelectMailItem	String – ID (140 Character ID)	String – Success	Selects Mail associated by the ID, see GetAllMailIDsArray() and similar Functions to get Array of Mail IDs.
GetAllMailIDsArray	String – Number Of Items	Array – Mail IDs	Returns Array containing IDs for Mails.
GetUnReadMailIDsArray	String – Number Of Items	Array – Mail IDs	Returns Array containing IDs for UnRead Mails.
GetReadMailIDsArray	String – Number Of Items	Array – Mail IDs	Returns Array containing IDs for Read Mails.
GetMailIDsArrayByFilter	String – Filter	Array – Mail IDs	Returns Array containing IDs for Mails satisfying Filter condition.
GetMailIDsArrayByDateRange	String – Start Date, String – End Date	Array – Mail IDs	Returns Array containing IDs for Mails received between given Start and End Date (Received Mail date in Inbox)
GetSenderSMTPAddress	N/A	String – Email Address	Returns the SMTP Email Address of Sender of selected Mail item.
GetSMTPAddressForCCRecipients Array	N/A	Array – Email Addresses	Returns Array containing SMTP Email Addresses of CC Recipients for the selected Mail item

GetSMTPAddressForToRecipients Array	N/A	Array – Email Addresses	Returns Array containing SMTP Email Addresses of To Recipients for the selected Mail item
GetNumberOfAttachments	N/A	String – No. of attachments	Returns number of attachments for selected Mail item
GetBody	N/A	String – Body	Returns the Body for selected Mail item in Plain Text format.
GetBodyHTML	N/A	String – Body	Returns the Body for selected Mail item in HTML format.
GetIsBodyHTML	N/A	String – “true” or “false”	Returns “true” if Body of selected Mail item is in HTML format, else “false”
GetSubject	N/A	String – Subject	Returns the subject for selected Mail item
GetSentOnDate	N/A	String – Date Sent	Returns the date selected Mail item was sent
GetIsUnReadStatus	N/A	String – “true” or “false”	Returns “true” if selected Mail item is Unread, else “false”
MarkAsRead	String – ID	String – Success	Changes the Unread status to Read for Mail item associated with given ID.
MarkAsUnRead	String – ID	String – Success	Changes the Unread status to Unread for Mail item associated with given ID.
DownloadAttachments	String – ID, String – DownloadAttachmentPath	String – Success	Downloads attachments to the path indicated for Mail item associated with given ID.

DisplayMail	String – ID	String – Success	Displays Mail associated with given ID.
CloseMail	String – ID	String – Success	Closes Mail associated with given ID.
DeleteMail	String – ID	String – Success	Deletes Mail associated with given ID.
MoveMail	String – ID, String – FolderPath (\\user.name@email.addr\Folder\SubFolder)	String – Success	Moves Mail associated with given ID to given FolderPath.
SendMail	String – ToRecipients (use Semi-colon), String – CcRecipients (use Semi-colon), String – BccRecipients (use Semi-colon), String – Subject, String – Attachments (use Semi-colon), String – Body, String – IsBodyHTML (pass "true" or "1" if Body is in HTML format), String - ReplyRecipients (use Semi-colon)	String – Success	Sends Mail using the selected Account. Use Semi-colon to separate multiple Recipients' SMTP Email Address and Attachment Paths. Body can be in Plain Text or HTML format. In case of HTML, pass "true" or "1" as value for *IsBodyHTML* input, else leave it blank.
SaveMailAsPDF	String – ID, String – FilePath (D:\folder1\folder2\filename.pdf)	String – Success	Saves Mail associated with given ID, as PDF Document using given FilePath.
GetFlagStatus	N/A	String – "Complete" or "Marked" or <blank>	Returns Flag Status as "Complete" or "Marked" and blank string in case of no Flag.
ClearMailFlag	String – ID	String – Success	Clears Flag for Mail associated with given ID.
MarkMailFlag	String – ID	String – Success	Marks Flag for Mail associated with given ID.
MarkMailFlagComplete	String – ID	String – Success	Marks Flag as Complete for Mail associated with given ID.

Meeting Requests

Function	Inputs	Outputs	Comments
SelectMeetingRequest	String – ID (140 Character ID)	String – Success	Selects Meeting Request associated by the ID, see GetMeetingRequestIDsArray() and similar Functions to get Array of Meeting Request IDs.
GetMeetingRequestIDsArray	String – Number Of Items	Array – Meeting Request IDs	Returns Array containing IDs for Meeting Requests.
GetMeetingRequestIDsArrayByFilter	String – Filter	Array – Meeting Request IDs	Returns Array containing IDs for Meeting Requests satisfying Filter condition.
GetMeetingRequestIDsArrayByDateRange	String – Start Date, String – End Date	Array – Meeting Request IDs	Returns Array containing IDs for Meeting Requests received between given Start and End Date.
GetMeetingRequestSenderSMTPAddress	N/A	String – Email Address	Returns the SMTP Email Address of Sender of selected Meeting Request.
GetMeetingRequestLocation	N/A	String – Location	Returns Location for the selected Meeting Request
GetMeetingRequestStartDateTime	N/A	String – Start Date Time	Returns Start Date Time for selected Meeting Request.
GetMeetingRequestEndDateTime	N/A	String – End Date Time	Returns End Date Time for selected Meeting Request.
GetMeetingRequestDurationInMinutes	N/A	String – Duration in Minutes	Returns Duration in Minutes for selected Meeting Request.
GetMeetingRequestNumberOfAttachments	N/A	String – No. of attachments	Returns number of attachments for selected Meeting Request
GetMeetingRequestBody	N/A	String – Body	Returns the body for selected Meeting Request
GetMeetingRequestSubject	N/A	String – Subject	Returns the subject for selected Meeting Request
GetMeetingRequestRecipientsArray	N/A	Array – Email Addresses	Returns Array containing SMTP Email Addresses of Recipients of selected Meeting Request.
GetMeetingRequestSentOnDate	N/A	String – Date Sent	Returns the date selected Meeting Request was sent

GetIsMeetingRequestUnRead	N/A	String – "true" or "false"	Returns "true" if selected Meeting Request is Unread, else "false"
MeetingRequestMarkAsRead	String – ID	String – Success	Changes the Unread status to Read for Meeting Request associated with given ID.
MeetingRequestMarkAsUnRead	String – ID	String – Success	Changes the Unread status to UnRead for Meeting Request associated with given ID.
DownloadMeetingRequestAttachments	String – ID, String – DownloadAttachmentPath	String – Success	Downloads attachments to the path indicated for Meeting Request associated with given ID.
DisplayMeetingRequest	String – ID	String – Success	Displays Meeting Request associated with given ID.
CloseMeetingRequest	String – ID	String – Success	Closes Meeting Request associated with given ID.
DeleteMeetingRequest	String – ID	String – Success	Deletes Meeting Request associated with given ID.
AcceptMeetingRequest	String – ID	String – Success	Accepts Meeting Request associated with given ID.
TentativeMeetingRequest	String – ID	String – Success	Chooses Tentative for Meeting Request associated with given ID.
DeclineMeetingRequest	String – ID	String – Success	Declines Meeting Request associated with given ID.
SendMeetingRequest	String – RequiredAttendees (use Semi-colon), String – OptionalAttendees (use Semi-colon), String – Subject, String – Location, String – StartDateTime, String – EndDateTime, String – Attachments (use Semi-colon), String – Body	String – Success	Sends Meeting Request using selected Account. Use Semi-colon to separate multiple Attendees' SMTP Email Address and Attachment Paths. Body can be in Plain Text only. Format for StartDateTime and EndDateTime: "01/17/2018 10:00 AM".

Appointments

Function	Inputs	Outputs	Comments
SelectAppointment	String – ID (140 Character ID)	String – Success	Selects Appointment associated by the ID, see GetAppointmentIDsArray( ) and similar Functions to get Array of Appointment IDs.
GetAppointmentIDsArray	String – Number Of Items	Array – Appointment IDs	Returns Array containing IDs for Appointments.
GetAppointmentIDsArrayByFilter	String – Filter	Array – Appointment IDs	Returns Array containing IDs for Appointments satisfying Filter condition.
GetAppointmentIDsArrayByDateRange	String – Start Date, String – End Date	Array – Appointment IDs	Returns Array containing IDs for Appointments that Start between given Start and End Date.
GetAppointmentOrganizerSMTPAddress	N/A	String – Email Address	Returns the SMTP Email Address of Organizer of selected Appointment.
GetAppointmentLocation	N/A	String – Location	Returns Location for the selected Appointment.
GetAppointmentStartDateTime	N/A	String – Start Date Time	Returns Start Date Time for selected Appointment.
GetAppointmentEndDateTime	N/A	String – End Date Time	Returns End Date Time for selected Appointment.
GetAppointmentDurationInMinutes	N/A	String – Duration in Minutes	Returns Duration in Minutes for selected Appointment.
GetAppointmentNumberOfAttachments	N/A	String – No. of attachments	Returns number of attachments for selected Appointment
GetAppointmentBody	N/A	String – Body	Returns the body for selected Appointment
GetAppointmentSubject	N/A	String – Subject	Returns the subject for selected Appointment
GetAppointmentRecipientsArray	N/A	Array – Email Addresses	Returns Array containing SMTP Email Addresses of Recipients/Attendees of selected Appointment.
DownloadAppointmentAttachments	String – ID, String – DownloadAttachmentPath	String – Success	Downloads attachments to the path indicated for Appointment associated with given ID.
DisplayAppointment	String – ID	String – Success	Displays Appointment associated with given ID.
CloseAppointment	String – ID	String – Success	Closes Appointment associated with given ID.
DeleteAppointment	String – ID	String – Success	Deletes Appointment associated with given ID.

CreateAppointment	String – Subject, String – Location, String – StartDateTime, String – EndDateTime, String – Attachments (use Semi-colon), String – Body	String – Success	Creates new Appointment on selected Calendar Folder (or default one). Use Semi-colon to separate multiple Attachment Paths. Body can be in Plain Text only. Format for StartDateTime and EndDateTime: "01/17/2018 10:00 AM".
GetIsAppointmentRecurring	N/A	String – "true" or "false"	Returns "true" if selected Appointment is part of Recurring Appointment Series, else "false".

Below are the common input variables:

Parameter Name	Type	Direction	Additional Info
vErrorFolder	String	Input	This is error folder inside bot folder which contains Logs and Snapshots folder (Already defined, no need to add value unless want to change location of file).
vLogFolder	String	Input	This folder contains Log file in case of error (Already defined, no need to add value unless want to change location of file).
vSnapshotFolder	String	Input	This folder contains Screenshot in case of error (Already defined, no need to add value unless want to change location of file).

Error Handling

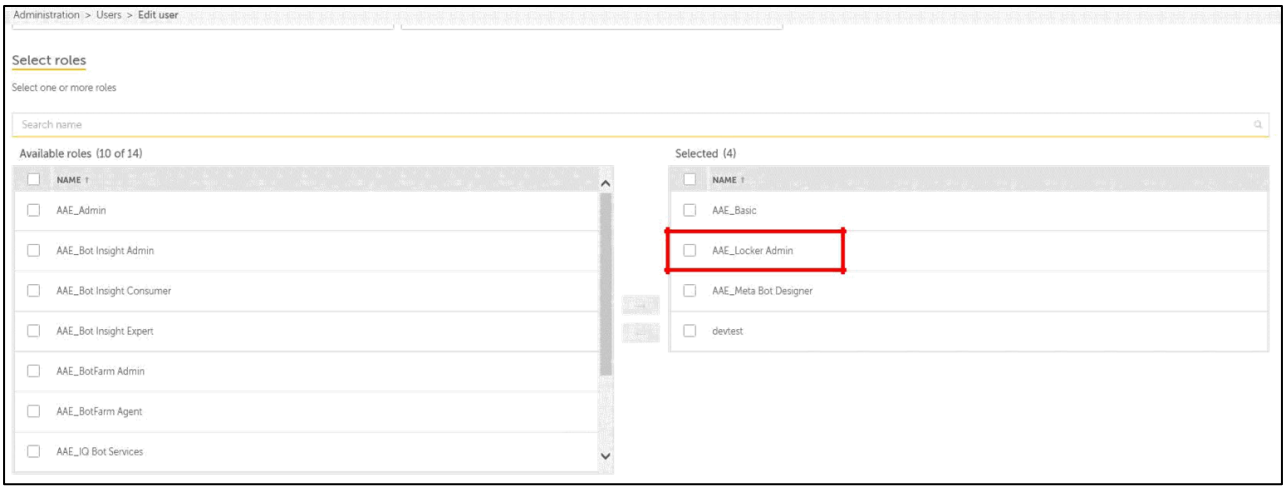
- Each Bot folder contains the belowhierarchy.
    - Error Folder
        - Logs
            - Error Logs Month-Day-Year.txt: In case of any error, this file logs error message along with time stamp
        - Snapshots
            - Error Snap Month-Day-Year HourMinuteSecond.png: In case of any error, this file captures screenshot of error.
- Task Status of bot is set to failed in case of error.

Note*

- It is possible for user having admin privileges to read and save the privileged files (open and write file functions), so user of the bot should not have admin access.

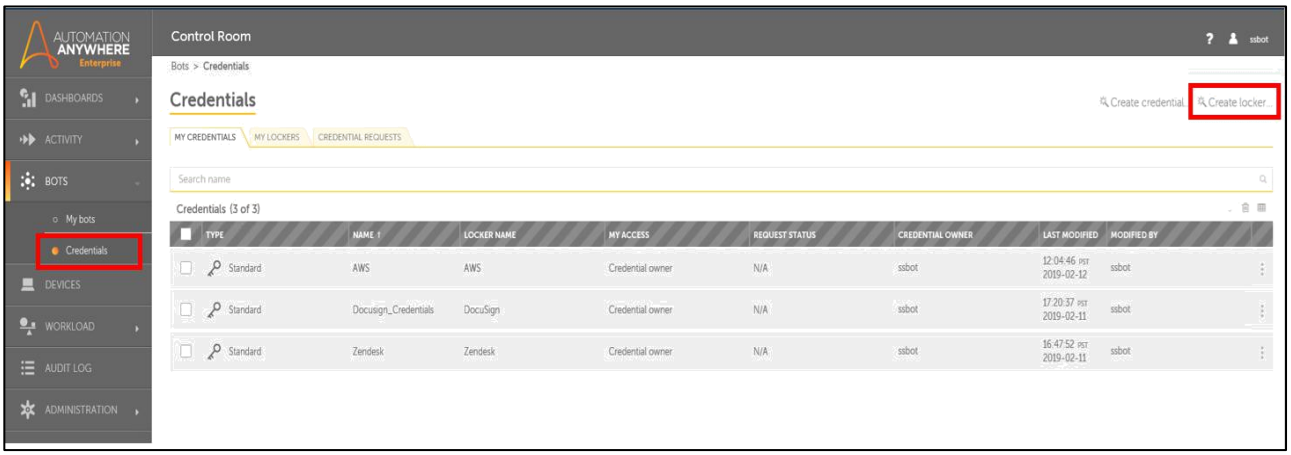
Steps to setup credential vault in Control Room

1. Log into control room as admin.
2. Go to users and assign "**AAE_Locker Admin**" role to user responsible for credential vault and save changes as shown below

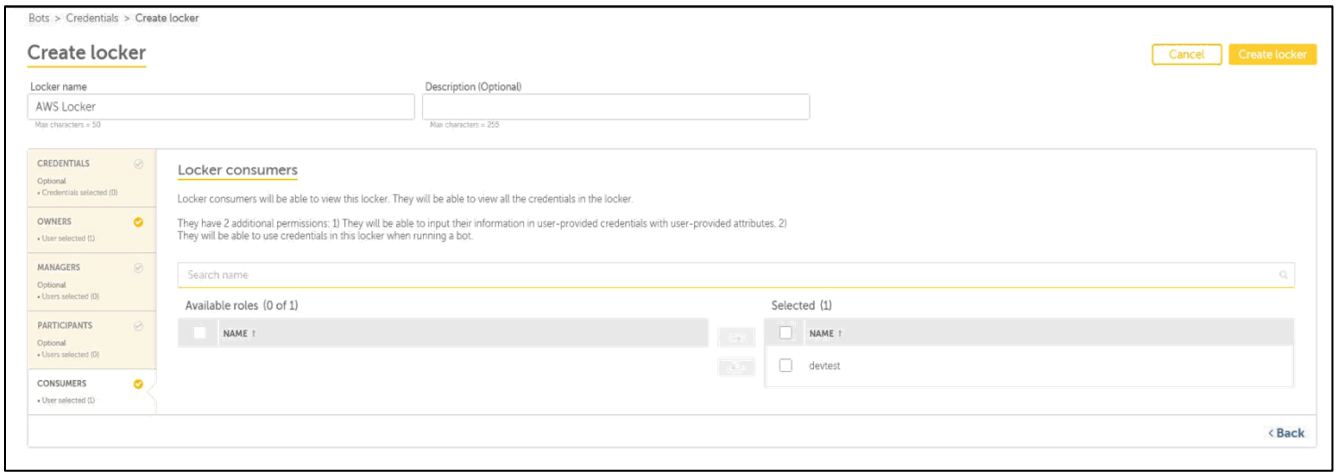


3. Now login in control room with user having "AAE_Locker Admin" role which is defined in above step.

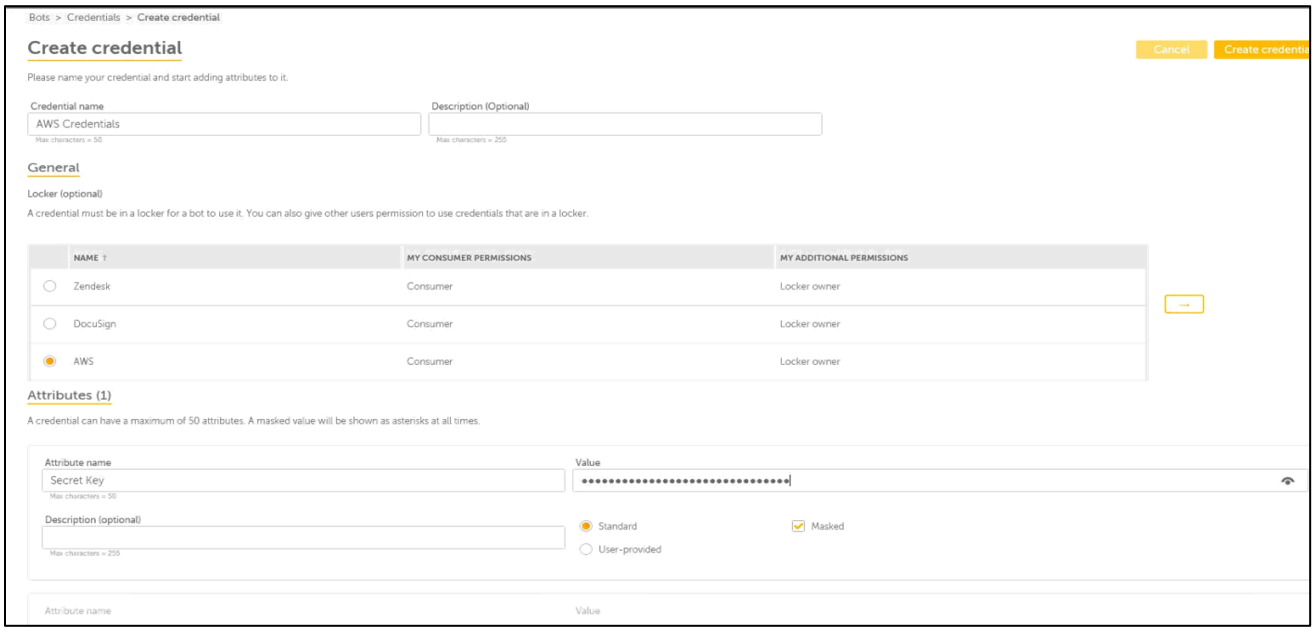
4. Navigate from left pane to Bots>Credentials and click "Create Locker" to create the locker for your credentials as shown below.



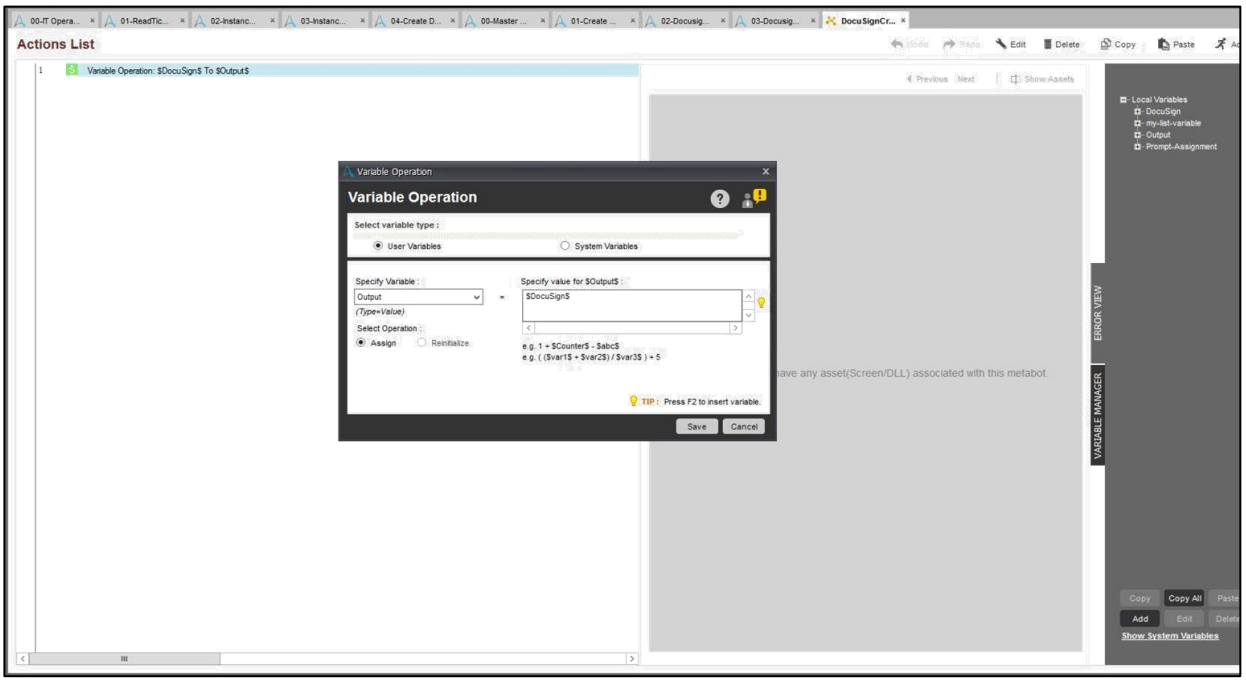
5. Provide locker name and provide all the options mainly consumers and click create locker.



6. Once the locker is created click create credentials. Provide credential name and assign locker created in above steps. Also provide attribute name and value as shown below. Check standard and if you want to hide the value mark it as masked. More attributes can be added by clicking add button.



7. These attributes can now be added directly as metabot variables or create a metabot and assign output to use in task as shown below.



Task Flow for Sample Tasks

The Outlook must be in a working condition before executing this task.
The below are the task bots for the sample use cases, no need of any configuration.

TaskBot flow for reading Mails

Variables to be filled –
Credential Name – SetSMTPAddress(Attribute Names - SetPR_SMTP_ADDRES, SetPR_SENT_REPRESENTING_ENTRYID) – Credential vault to be configured.
vNumberOfItems(Task Bot)

1. LaunchOutlook (optional – call only if you want Outlook Application Window visible)
2. GetSmtpAddressURL (to be passed from Control Room). Please refer to point 9 above to get the values. Credential vault needs to be configured and the values should flow from Crendential vault to Task Bots.
3. SelectProfile or SelectProfileByNamePassword
4. SelectAccount or SelectAccountByName
5. SelectInbox or SelectFolder
6. GetMailIDsArray or alternate Function (vNumberOfItems variable value to be filled)
7. Loop to iterate over Mail IDs and process them.

TaskBot flow for reading Appointments

Variables to be filled –

Credential Name – SetSMTPAddress(Attribute Names - SetPR_SMTP_ADDRES, SetPR_SENT_REPRESENTING_ENTRYID) – Credential vault to be configured.

vStartDate & vEndDate (Task Bot)

1. LaunchOutlook (optional – call only if you want Outlook Application Window visible)
2. GetSmtpAddressURL (to be passed from Control Room). Please refer to point 9 above to get the values. Credential vault needs to be configured and the values should flow from Crendential vault to Task Bots.
3. SelectProfile or SelectProfileByNamePassword
4. SelectAccount or SelectAccountByName
5. SelectCalendar
6. GetAppointmentIDsArrayByDateRange or alternate Function (vStartDate & vEndDate variable values to be mentioned)
7. Loop to iterate over Appointment IDs, process them and display.

TaskBot flow for reading and processing Meeting Requests

Variables to be filled –
Credential Name – SetSMTPAddress(Attribute Names - SetPR_SMTP_ADDRES, SetPR_SENT_REPRESENTING_ENTRYID) – Credential vault to be configured.

vStartDate & vEndDate (Task Bot)

Disclaimer - This bot accepts a meeting within the mentioned date range if the duration is less than 60 mins, else declines it.
Disable sample code line 22-26 if not required.

1. LaunchOutlook (optional – call only if you want Outlook Application Window visible)
2. GetSmtpAddressURL (to be passed from Control Room). Please refer to point 9 above to get the values. Credential vault needs to be configured and the values should flow from Crendential vault to Task Bots.
3. SelectProfile or SelectProfileByNamePassword
4. SelectAccount or SelectAccountByName
5. SelectInbox or SelectFolder
6. GetMeetingRequestIDsArrayByDateRange or alternate Function (vStartDate & vEndDate variable values to be mentioned)

7. Loop to iterate over Meeting Request IDs, process them and display.

TaskBot flow for Saving mail as PDF

Variables to be filled –
Credential Name – SetSMTPAddress(Attribute Names - SetPR_SMTP_ADDRES, SetPR_SENT_REPRESENTING_ENTRYID) – Credential vault to be configured.

vNumberOfItems, vFilepath (Task Bot)

1. LaunchOutlook (optional – call only if you want Outlook Application Window visible)
2. GetSmtpAddressURL (to be passed from Control Room). Please refer to point 9 above to get the values. Credential vault needs to be configured and the values should flow from Crendential vault to Task Bots.
3. SelectProfile or SelectProfileByNamePassword
4. SelectAccount or SelectAccountByName
5. SelectInbox or SelectFolder
6. GetMailIDsArrayByDateRange or alternate Function
7. Loop to iterate over Appointment IDs and process them.

TaskBot flow for Sending a sample Mail

This bot takes the existing account and sends a mail to self. Customizable on changing the variable values and variable operations.

Variables to be filled –
Credential Name – SetSMTPAddress(Attribute Names - SetPR_SMTP_ADDRES, SetPR_SENT_REPRESENTING_ENTRYID) – Credential vault to be configured.

1. LaunchOutlook (optional – call only if you want Outlook Application Window visible)
2. GetSmtpAddressURL (to be passed from Control Room). Please refer to point 9 above to get the values. Credential vault needs to be configured and the values should flow from Crendential vault to Task Bots.
3. SelectProfile or SelectProfileByNamePassword
4. SelectAccount or SelectAccountByName
5. Get Account Information for fetching the SMTP mail ID
6. String Manipulation for filtering the mail ID (recipient)
7. Send Email

For more implementation doubts, visit the link below -
Apeople Community

Additional Resources

Automation Anywhere provides a Product Documentation portal (https://docs.automationanywhere.com/) which you can access for more information about our products and building bots and Digital Workers.

The "Build" section of the portal includes these sections:

- Getting Started - information on building bots and recommended practices (including use of the Credential Vault)
- Build Advanced Bots - details on Metabots and the approach to integrating code into them
- Build Digital Workers - high level architecture

Questions on bot development can also be posted to our Community site: https://apeople.automationanywhere.com.