

## Perform Various Email Operations-Automation Anywhere-Readme

**Latest Version – July 2020 (Update Notes [below](#)):**

### **Bot Overview**

This bot performs the following:

- Performs several Microsoft Outlook Operations. Description of all the operations can be found in this document. The Latest release notes can also be found below.

### **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

**Latest Update Notes (July 2020):**

✓ Addition of the below functions:

Sl no.	Function	Input	Output	Comments
1	SelectFolderByPath	String - Folder Path of the Outlook Ex: \\xyz@automationanywhere.com\Sent Items	Selects the target folder, Outputs Success on successful execution	Required for special cases like selecting folders and performing next actions like saving attachment, reading etc.
2	GetIsBodyRTF	NA	True or False	Verifies whether a mail body contains Rich Text Formatted text and returns a Boolean.
3	GetListOfAttachments	NA	Files names Ex: A;B;C Where A,B,C are file names	Returns the Files name/names separated by; Returns blank if no attachments are found.
4	SaveMailasMsg	String - Download Path and Array - Mail ID	Success on successful download	Downloads the mail as Outlook message in the target folder. Ex: D:\test\A The extension will be auto applied.
5	SaveMailasTxt	String - Download Path and Array - Mail ID	Success on successful download	Downloads the mail as Text file in the target folder. Ex: D:\test\A The extension will be auto applied.
6	SaveMailasHTML	String - Download Path and Array - Mail ID	Success on successful download	Downloads the mail as HTML in the target folder along with the HTML files in separate folder within the same target folder. Ex: D:\test\A The extension will be auto applied.
7	ComposeNewMail	NA	Outputs success and opens the Compose New Outlook Window	This activity is mainly for attended automation where the Outlook Window opens for certain actions to be performed.
8	ComposeReplyMail	Array - Mail ID	Outputs success and opens the Reply Outlook Window against the Mail ID to reply to the sender	This activity is mainly for attended automation where the Outlook Window opens for replying to certain Mail.
9	composeReplyAllMail	Array - Mail ID	Outputs success and opens the Reply Outlook Window against the Mail ID to reply all	This activity is mainly for attended automation where the Outlook Window opens for replying all participants in the mail.
10	composeForwardMail	Array - Mail ID	Outputs success and opens the Forward Outlook Window against the Mail ID to reply all	This activity is mainly for attended automation where the Outlook Window opens for forwarding the mail.

11	ReplyMail	String - Reply Recipient, CC, BCC, Subject, Attachments, Body, Array - Mail ID	Outputs Success and sends replies to the sender in background	This will only reply to the sender, Mail Participants can be put in CC, BCC as well. CC, BCC are optional and can be feeded as Blank inputs as null variables on requirement.
12	ReplyAllMail	String - Reply Recipient, CC, To, BCC, Subject, Attachments, Body, Array - Mail ID	Outputs Success and sends replies to all the participants in mail in background	This will reply all, Mail Participants can be put in CC, BCC as well. CC, BCC are optional and can be feeded as Blank inputs as null variables on requirement. Note: It will only send to persons who are part of the mail that is being replied to.
13	ForwardMail	String - Reply Recipient, CC, To, BCC, Subject, Attachments, Body, Array - Mail ID	Outputs Success and forwards specific mail to single/multiple participants in background	This will forward Mail to single/multiple participants as feeded in input. Note: For adding multiple participants, use the separator ; Ex: A;B;C Where A, B, C are email IDs
14	GetSMTPAddressForRecipientsString	NA	String - vOutput	Outputs strings of participants in To, Ex- A;B;C where A,B,C are email IDs
15	GetSMTPAddressCCRecipientsString	NA	String - vOutput	Outputs strings of participants in CC, Ex- A;B;C where A,B,C are email IDs
16	GetImportance	NA	String - vOutput	Outputs the importance of Mail Items (High, Normal, Medium)
17	GetVotingResponse	NA	String - vOutput	Outputs Yes/No as voting response. This is only applicable for Voting responses.
18	GetReadReceiptRequested	NA	String - vOutput	Outputs True/False according to Read receipt request
19	GetReceivedTime	NA	String - vOutput	Outputs the DateTime of the mail receive date
20	GetDownloadState	NA	String - vOutput	Outputs the download state of emails. Ex- Ol FullItem, Incomplete etc.

✓ **Note: “Object reference not set to an instance of object” – If this Error appears, please follow the below steps:**

1. Close all apps
2. Restart the system/Virtual Machine wherever is the platform installed.
3. Open Temp and delete all temp files (Click ‘Win+R’ button and delete all files)
4. Open the application and try again.

✓ [Troubleshooting Guides Below](#)

✓ [Adding DASL Outlook Query to loop through specific target subject Unread Mails without changing the status of other unread mails](#)

## 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 Callingcorresponding 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. **Adding DASL Queries:** You can also use some functions for reading all Unread Emails implementing DASL Queries that would return all unread/read queries and further implementing Array Loops around it to go till the last unread mail with specific subjects avoiding the need to touch other unread mails  
 Ex: This DASL Query returns all UnRead Mail IDs (@SQL= urn:schemas:httpmail:read=0)  
 Use this by applying commands like:  
 GetMailIDsArrayByFilter, by putting the Filter as the above query and output to catch all the results of the query. Further by Array Looping those unread IDs, gives you many desired results.

```

9  Run Logic "SelectInbox" from MetaBot "My MetaBots\OutlookV2.mbot"
10 IF $vOutput$ Equal To (=) "Success" Then
11   Run Logic "GetMailIDsArrayByFilter" from MetaBot "My MetaBots\OutlookV2.mbot"
12   Start Loop "$ArrayRows($arrUnreadMailIDArray)$" Times
13     Comment: Get unread email count
14     Variable Operation: $Counter$ To $vUnreadMailCount$
15   End Loop
16   IF $vUnreadMailCount$ Not Equal To (<>) "0" Then
17     Start Loop "$ArrayRows($arrUnreadMailIDArray)$" Times
18       Comment: Read each unread email and check email subject
19       Variable Operation: $arrUnreadMailIDArray($Counter$.1)$ To $vCurrentMailID$
20       Run Logic "SelectMailItem" from MetaBot "My MetaBots\OutlookV2.mbot"
21       Run Logic "GetSubject" from MetaBot "My MetaBots\OutlookV2.mbot"
22       IF $vEmailSubject$ Includes "TargetSubject" Then
23         Variable Operation: $Counter$ To $vReadMailCount$
24         Comment: Downloading attachment if Subject match
25         Run Logic "DownloadAttachments" from MetaBot "My MetaBots\OutlookV2.mbot"
26         IF $vDownloadAttachmentLogicOutput$ Equal To (=) "Success" Then
27           Comment: Download attachment and mark as read
28           Run Logic "MarkAsRead" from MetaBot "My MetaBots\OutlookV2.mbot"
29         ELSE

```

12. 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.
13. 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.
14. 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".
15. 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](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.



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

## 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.
GetAccountInformation	N/A	String – Account Information	Returns: DisplayName, UserName, SmtptAddress, 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- “ <a href="http://schemas.microsoft.com/ma&lt;br/&gt;pi/proptag/0x39FE001E">http://schemas.microsoft.com/ma pi/proptag/0x39FE001E</a> ” The value of SetPR_SENT_REPRESENTING_ENTRY ID - “ <a href="http://schemas.microsoft.com/ma&lt;br/&gt;pi/proptag/0x00410102">http://schemas.microsoft.com/ma pi/proptag/0x00410102</a> ”	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 attachmen ts	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 <i>IsBodyHTML</i> 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.
GetMeetingRequestSenderSMTPAddresses	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

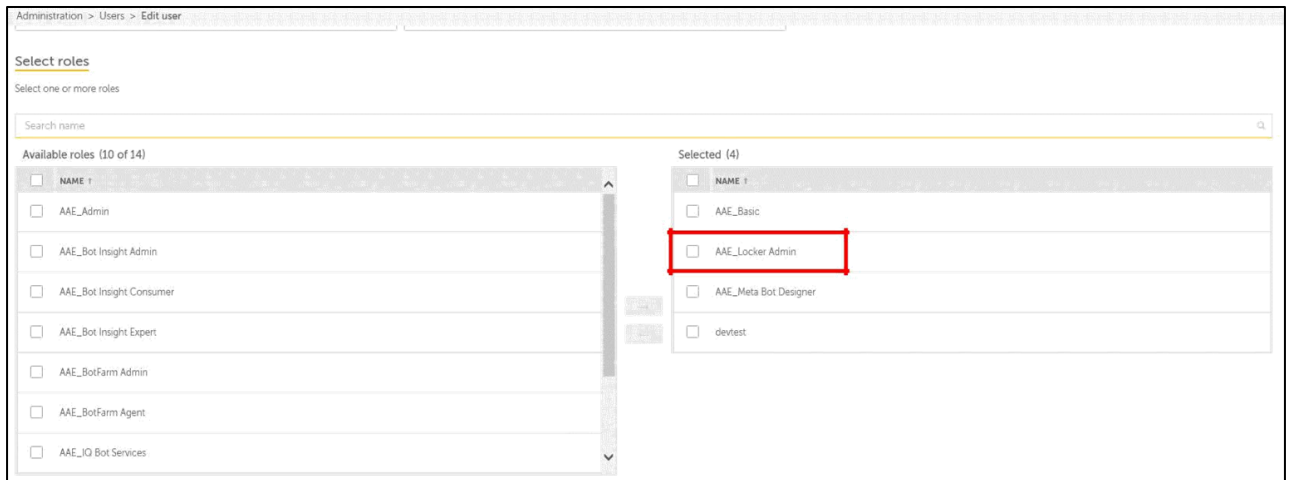
- Each Bot folder contains the below hierarchy.
  - o Error Folder
    - Logs
      - Error Logs Month-Day-Year.txt: In case of any error, this file logs error message along with time stamp
    - Snapshots
      - Error Snap Month-Day-Year HourMinuteSecond.png: In case of any error, this file captures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

### Note\*

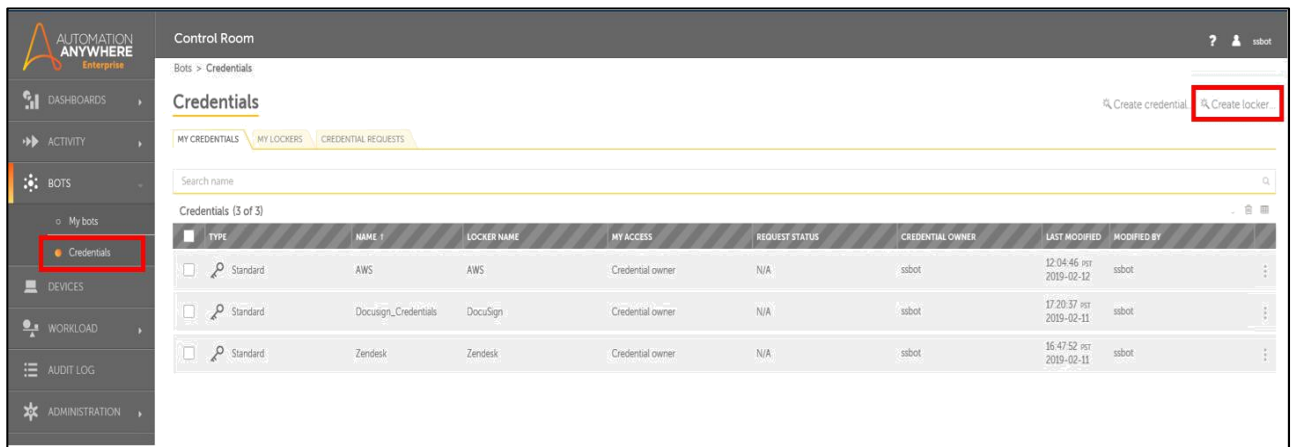
- It is possible for user having admin privileges to read and save the privileged files (open and write file functions), so user of the bot should not have admin access.

## 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.

Bots > Credentials > Create locker

### Create locker

[Cancel](#) [Create locker](#)

Locker name:  (Max characters = 50) Description (Optional):  (Max characters = 255)

**CREDENTIALS** ☒ Optional  
• Credentials selected (0)

**OWNERS** ☒ • User selected (0)

**MANAGERS** ☒ • Users selected (0)

**PARTICIPANTS** ☒ • Users selected (0)

**CONSUMERS** ☒ • User selected (0)

#### Locker consumers

Locker consumers will be able to view this locker. They will be able to view all the credentials in the locker.

They have 2 additional permissions: 1) They will be able to input their information in user-provided credentials with user-provided attributes. 2) They will be able to use credentials in this locker when running a bot.

Search name:

Available roles (0 of 1)

<input type="checkbox"/>	NAME
<input type="checkbox"/>	NAME

Selected (1)

<input type="checkbox"/>	NAME
<input type="checkbox"/>	devtest

[< Back](#)

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.

Bots > Credentials > Create credential

### Create credential

[Cancel](#) [Create credential](#)

Please name your credential and start adding attributes to it.

Credential name:  (Max characters = 50) Description (Optional):  (Max characters = 255)

#### General

Locker (optional)

A credential must be in a locker for a bot to use it. You can also give other users permission to use credentials that are in a locker.

NAME	MY CONSUMER PERMISSIONS	MY ADDITIONAL PERMISSIONS
<input type="radio"/> Zendesk	Consumer	Locker owner
<input type="radio"/> DocuSign	Consumer	Locker owner
<input checked="" type="radio"/> AWS	Consumer	Locker owner

[Add](#)

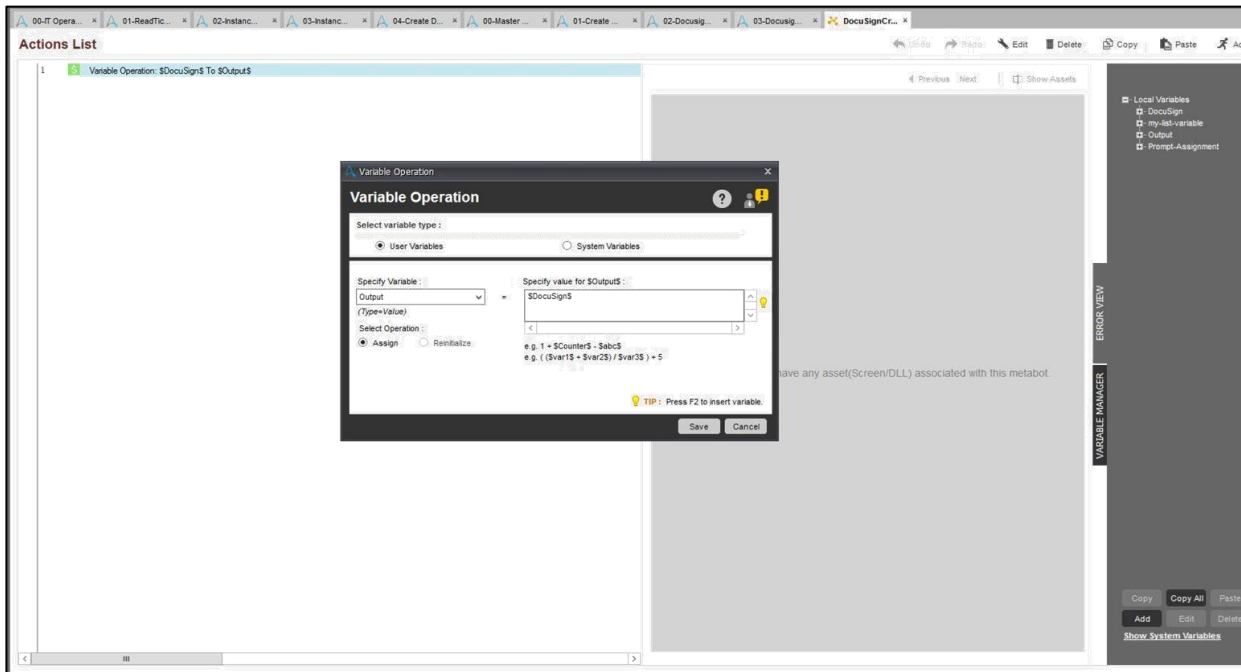
#### Attributes (1)

A credential can have a maximum of 50 attributes. A masked value will be shown as asterisks at all times.

Attribute name: <input type="text" value="Secret Key"/> (Max characters = 50)	Value: <input type="text" value="*****"/> <a href="#">Show</a>
Description (optional): <input type="text"/> (Max characters = 255)	<input checked="" type="radio"/> Standard <input type="radio"/> User-provided <input checked="" type="checkbox"/> Masked

Attribute name	Value
----------------	-------

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 Credential 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 Credential 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 Credential 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 Credential 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 Credential 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>.

### **References**

No.	Topic	Reference Link
1	System Requirements - Client	<a href="#">Click here</a>
2	System Requirements – Control Room	<a href="#">Click here</a>
3	How to create credential & Locker?	<a href="#">Click here</a>
4	Credential Vault Setup Guide	<a href="#">Click here</a>
5	Apeople Community Site	<a href="#">Click here</a>
6	Product Documentation portal	<a href="#">Click here</a>
7	Perform Various Outlook Operation - Botstore	<a href="#">Click here</a>
8	Common Troubleshooting ways	<a href="#">Click here</a>
9	Public Botstore Website	<a href="#">Click here</a>
10	Oracle Taleo Fluid Recruiting App	<a href="#">Click here</a>