INTEGRATION PACK FOR OAUTH EMAIL

For Microsoft System Center Orchestrator

For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_OAuth_Email_2.0**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_OAuth_Email_x64_2.0**

User Guide

Version 2.0

Kelverion Integration Pack for OAuth Email

Published: January 2023

Feedback

Send suggestions and comments about this document to support@kelverion.com

Contents

Kelverion Integration Pack for OAuth Email 2

Contents 3

Installation and Configuration 4
- System Requirements 4
- Registering and Deploying the Integration Pack 5
- Licensing the Integration Pack 6
- Configuring the Kelverion Integration Pack for OAuth Email 6
- Connecting to Microsoft Office 365 11
  - Register Azure AD Application 11
  - Grant Exchange Online Permissions 14
  - Enable SMTP in Exchange Online 14
- Message Identification 15
  - Considerations When Using IMAP 15

Email Activities 16
- Common Configuration Instructions for All Activities 16
  - Activity Properties 16
  - General Tab 16
  - Properties/Filters Tab 16
  - Run Behavior Tab 17
  - Published Data 18

Delete Email Activity 20

Empty Mailbox Activity 21

Get Email Activity 22

Get Email List Activity 24

Get Mailbox Info Activity 27

Get Mailbox List Activity 28

Monitor Email Activity 29

Move Email Activity 31

Send Email Activity 32

Installation and Configuration

The Integration Pack for OAuth Email is an add-on for System Center Orchestrator that enables you to send, retrieve and monitor email messages with email accounts in your Microsoft Office 365 Subscription. The IP supports the SMTP (outgoing email) and IMAP (incoming email) protocols.

System Requirements

The Kelverion Integration Pack for OAuth Email requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring System Center Orchestrator refer to the respective product documentation.

Kelverion_Integration_Pack_for_OAuth_Email (32-bit)

- Microsoft System Center Orchestrator *2016, 2019*
- Microsoft .NET Framework 4.7.2
- Microsoft Office 365 Email Subscription

Kelverion_IP_OAuth_Email_x64 (64-bit)

- Microsoft System Center Orchestrator 2022
- Microsoft .NET Framework 4.7.2
- Microsoft Office 365 Email Subscription

The Integration Pack requires the following protocols to be enabled in your Microsoft Office 365 Email Subscription:

- SMTP for Send Email activity
- IMAP for all other activities

Important: The IP supports **OAuth 2.0** authentication for the **Microsoft Office 365** environment. You will need to register and configure an app in your Azure Active Directory, that the IP will use for accessing your Office 365 email account(s).

Important: For incoming email, the IP supports **IMAP,** but does not support the POP3. For outgoing email, the IP supports **SMTP**.

Important: The IP supports **Basic Authentication** for email providers that still support it, for example Outlook.com (Hotmail). The Integration Pack should be compatible with standard **IMAP** and **SMTP** implementations, however in cases where server implementations differ from the standard, the IP may not behave as expected.

Registering and Deploying the Integration Pack

After you download the integration pack, you register the integration pack file with the Orchestrator management server, and then deploy it to runbook servers and computers that have the Runbook Designer installed.

IMPORTANT: Ensure that you are deploying the correct version of the Integration Pack.

- For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_OAuth_Email**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_OAuth_Email_x64**

To register the integration pack:

1. On the management server, copy the **.OIP** file for the integration pack to a local hard drive or network share.
2. Confirm that the file is not set to **Read Only** to prevent unregistering the integration pack later.
3. Start the **Deployment Manager**.
4. In the navigation pane of the Deployment Manager, expand **Orchestrator Management Server**, right-click **Integration Packs** to select **Register IP with the Orchestrator Management Server**. The **Integration Pack Registration Wizard** opens.
5. Click **Next**.
6. In the **Select Integration Packs or Hotfixes** dialog box, click **Add**.
7. Locate the **.OIP** file that you copied locally from step 1, click **Open** and then click **Next**.
8. In the **Completing the Integration Pack Wizard** dialog box, click **Finish**.
9. On the **End User Agreement** dialog box, read the Kelverion License Terms**,** and then click **Accept**.
10. The **Log Entries** pane displays a confirmation message when the integration pack is successfully registered.

To deploy the integration pack:

1. In the navigation pane of the **Deployment Manager**, right-click **Integration Packs**, click **Deploy IP to Runbook Server or Runbook**.
2. Select the integration pack that you want to deploy, and then click **Next**.
3. Enter the name of the runbook server or computers with the Runbook Designer installed, on which you want to deploy the integration pack, click **Add**, and then click **Next**.
4. Continue to add additional runbook servers and computers running the Runbook Designer, on which you want to deploy the integration pack. Click **Next**.
5. In the **Installation Options** dialog box configure the following settings.

6. To choose a time to deploy the integration pack, select the **Schedule installation** check box, and then select the time and date from the **Perform installation** list.
7. Click one of the following:
    a. **Stop all running runbooks before installing the integration pack** to stop all running runbooks before deploying the integration pack.
    b. **Install the Integration Packs without stopping the running Runbooks** to install the integration pack without stopping any running runbooks.
8. Click **Next**.
9. In the **Completing Integration Pack Deployment Wizard** dialog box, Click **Finish**.
10. When the integration pack is deployed, the **Log Entries** pane displays a confirmation message.

For more information about how to install integration packs, see the [How to Install an Integration Pack](https://technet.microsoft.com/en-us/library/hh420346.aspx) (https://technet.microsoft.com/en-us/library/hh420346.aspx).

Licensing the Integration Pack

After you register and deploy the integration pack, you must provide a valid Kelverion license before running any runbooks that contain activities from the integration pack.

To deploy the integration pack license file to System Center Orchestrator 2019 or earlier:

1. Copy the .KAL license file to %PROGRAMFILES(X86)%\Kelverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

To deploy the integration pack license file to System Center Orchestrator 2022 or later:

1. Copy the .KAL license file to %PROGRAMFILES%\Kelverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

Configuring the Kelverion Integration Pack for OAuth Email

A configuration establishes a reusable link between Orchestrator and the target email server. You can create multiple configurations, as you require, specifying links to multiple email servers. You can also create multiple configurations to the same email server to allow for different user accounts. For details on Microsoft Office 365 email connection settings, please refer to https://support.microsoft.com/en-us/office/pop-imap-and-smtp-settings-8361e398-8af4-4e97-b147-6c6c4ac95353.

To set up Email Configuration for OAuth authentication (Office 365):

1. In the Runbook Designer, click the **Options** menu, and select *KA OAuth Email*. The **KA OAuth Email** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.

3. In the **Name** box, enter a name for the configuration. This could be the name of the email server or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (…) next to the **Type** box and select ***Email Configuration***.
5. In the **Email Address** box, type the email address associated with the email account. This address will be used when sending emails. When connecting to Microsoft Office 365, you will have to grant access for your Azure AD registered application to access this email account. For more details, please refer to Connecting to Microsoft Office 365.
6. In the **Incoming Protocol** box select **IMAP** to indicate the type of email server used to retrieve email. POP3 is not supported.
7. In the **Incoming Server** box, enter the FQDN or the IP address of the incoming email server. By default, this property is set to **outlook.office365.com**, which currently is the Microsoft Office 365 IMAP server.
8. In the **Incoming Server Port** box enter the port used to communicate with the email server. By default, this property is set to **993**, which currently is the Microsoft Office 365 IMAP port.
9. In the **Incoming Tenant ID** box, type the tenant ID for your Microsoft Office 365 subscription. For details, please see Connecting to Microsoft Office 365.
10. In the **Incoming Client ID** box, type the client ID of the Azure AD app that you have configured to connect to your Microsoft Office 365 subscription. For details, please see Connecting to Microsoft Office 365.
11. In the **Incoming Client Secret** box, type the client secret for the Azure AD app that you have configured to connect to your Microsoft Office 365 subscription. Note that the **Incoming User Password** is ignored when **Incoming Client Secret** is configured. For details, please see Connecting to Microsoft Office 365.
12. In the **Incoming User Name** box, type the user name or email address used to connect to the incoming server.
13. In the **Incoming User Password** box, type the password of the user that is used to connect to the incoming server. Note that **Incoming User Password** is ignored when **Incoming Client Secret** is configured. If you intend to connect with user name and password, make sure to leave **Incoming Client Secret** empty.
14. In the **Incoming Timeout (Seconds)** box, type the number of seconds after which an email operation should time out.
15. In the **Incoming Retries** box, type the number of times the IP should retry a failed mail operation. Type zero to disable retries.
16. In the **Incoming Retry Delay (Seconds)** box, type the number of seconds the IP should wait between retry attempts.
17. In the **Outgoing Server (SMTP)** box, type the FQDN or the IP address of the outgoing email server. By default, this property is set to **smtp.office365.com**, which currently is the Microsoft Office 365 SMTP server. When sending email with Microsoft Office 365, you will have to enable SMTP for the configured **Email Address**. For more details, please refer to Connecting to Microsoft Office 365.

18. In the **Outgoing Server Port** box, type the port used to communicate with the outgoing server. By default, this property is set to **587**, which currently is the Microsoft Office 365 SMTP port.

19. In the **Outgoing Uses Different Credentials** box**,** indicate whether the outgoing server uses different credentials than the incoming server. When set to **False**, the **Outgoing Tenant ID, Client ID, User Name** and **User Password** properties are ignored, and the IP will use the corresponding **Incoming** settings.

    Note: The **User Password** is required for **Outgoing** settings, so if the password is not specified in your **Incoming** settings, you will have to use separate **Outgoing** settings.

20. In the **Outgoing Tenant ID** box, type the tenant ID for your Microsoft Office 365 subscription. For details, please see Connecting to Microsoft Office 365.

21. In the **Outgoing Client ID** box, type the client ID of the Azure AD app that you have configured to connect to your Microsoft Office 365 subscription. For details, please see Connecting to Microsoft Office 365.

22. In the **Outgoing User Name** box, type the user name or email address used to connect to the SMTP server.

23. In the **Outgoing User Password** box, type the password for of the user used to connect to the SMTP server.

24. In the **Outgoing Timeout (Seconds)** box, enter the number of seconds after which an email operation should time out.

25. In the **Outgoing Retries** box, enter the number of times the IP should retry a failed remote operation. Type zero to disable retries.

26. In the **Outgoing Retry Delay (Seconds)** box enter the number of seconds to wait between retry attempts.

27. In the **Trash Folders** box, enter a comma (,) separated list of folder names that the IP should treat as trash folders. This list is used by the Delete Email Activity to determine how email messages should be deleted from a mailbox.

28. Click **OK** to close the configuration dialog box, and then click **Finish**.

To set up Email Configuration for Basic Authentication (Outlook.com, Hotmail):

1. In the Runbook Designer, click the **Options** menu, and select *KA OAuth Email*. The **KA OAuth Email** dialog box appears.

2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.

3. In the **Name** box, enter a name for the configuration. This could be the name of the email server or a descriptive name to distinguish the type of configuration.

4. Click the ellipsis button (…) next to the **Type** box and select ***Email Configuration***.

5. In the **Email Address** box, type the email address associated with the email account. This address will be used when sending emails.

6. In the **Incoming Protocol** box select **IMAP** to indicate the type of email server used to retrieve email. POP3 is not supported.
7. In the **Incoming Server** box, enter the FQDN or the IP address of the incoming email server. For **Outlook.com** or **Hotmail**, set this property to **imap-mail.outlook.com.**
8. In the **Incoming Server Port** box enter the port used to communicate with the email server. By default, this property is set to **993**, which is the default IMAP port.
9. Leave the **Incoming Tenant ID** box empty.
10. Leave the **Incoming Client ID** box empty.
11. Leave the **Incoming Client Secret** box empty.
12. In the **Incoming User Name** box, type the user name or email address used to connect to the incoming server.
13. In the **Incoming User Password** box, type the password of the user that is used to connect to the incoming server.
14. In the **Incoming Timeout (Seconds)** box, type the number of seconds after which an email operation should time out.
15. In the **Incoming Retries** box, type the number of times the IP should retry a failed mail operation. Type zero to disable retries.
16. In the **Incoming Retry Delay (Seconds)** box, type the number of seconds the IP should wait between retry attempts.
17. In the **Outgoing Server (SMTP)** box, type the FQDN or the IP address of the outgoing email server. For **Outlook.com** or **Hotmail**, set this property to **smtp-mail.outlook.com.**
18. In the **Outgoing Server Port** box, type the port used to communicate with the outgoing server. By default, this property is set to **587**, which is the default SMTP port.
19. In the **Outgoing Uses Different Credentials** box**,** indicate whether the outgoing server uses different credentials than the incoming server. When set to **False**, the **Outgoing User Name** and **Outgoing User Password** properties are ignored, and the IP will use the corresponding **Incoming** settings.

    Note: The **User Password** is required for **Outgoing** settings, so if the password is not specified in your **Incoming** settings, you will have to use separate **Outgoing** settings.
20. Leave the **Outgoing Tenant ID** box empty.
21. Leave the **Outgoing Client ID** box empty.
22. In the **Outgoing User Name** box, type the user name or email address used to connect to the SMTP server.
23. In the **Outgoing User Password** box, type the password for of the user used to connect to the SMTP server.
24. In the **Outgoing Timeout (Seconds)** box, enter the number of seconds after which an email operation should time out.

25. In the **Outgoing Retries** box, enter the number of times the IP should retry a failed remote operation. Type zero to disable retries.
26. In the **Outgoing Retry Delay (Seconds)** box enter the number of seconds to wait between retry attempts.
27. In the **Trash Folders** box, enter a comma (,) separated list of folder names that the IP should treat as trash folders. This list is used by the Delete Email Activity to determine how email messages should be deleted from a mailbox.

Click **OK** to close the configuration dialog box, and then click **Finish**.

Tip: When building new email configurations, it is recommended to reduce Timeout and disable Retries, so that configuration errors can be quickly diagnosed. Once connection is established and IP communicates successfully with the email server, then Timeout and Retries should be tuned appropriately.

Connecting to Microsoft Office 365

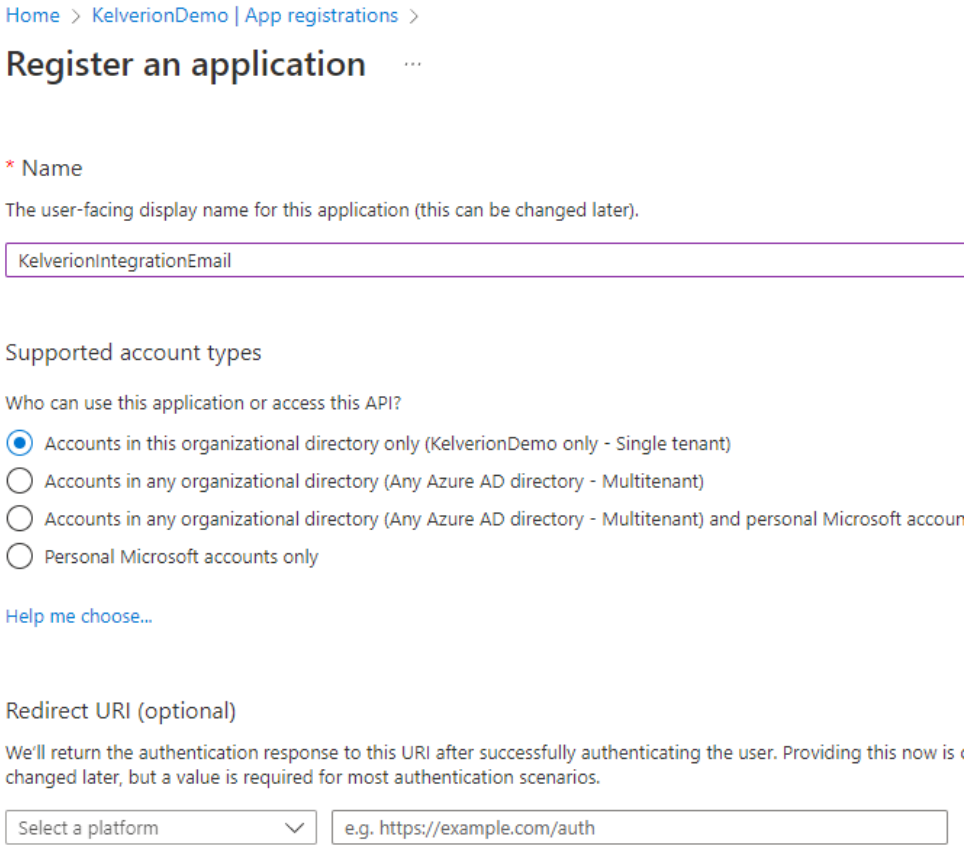
When using the Integration Pack with Microsoft Office 365, before sending or reading email, several configuration steps are required in your Office 365 Azure and Exchange environment.

- Register Azure AD Application – used by the IP to connect using OAuth.
- Grant Exchange Online Permissions – permissions for the configured email account.
- Enable SMTP in Exchange Online – enable SMTP for the configured email account.

Register Azure AD Application

When using OAuth to connect to Microsoft Office 365, an application must be registered with Azure Active Directory. Please follow the steps bellow to create and configure an Azure AD application. For additional details, you can refer to the Microsoft Documentation https://learn.microsoft.com/en-us/exchange/client-developer/legacy-protocols/how-to-authenticate-an-imap-pop-smtp-application-by-using-oauth.

1. Log into the Azure Portal for your Office 365 Subscription, using a tenant admin user.
2. In Azure Active Directory, under **App registrations**, create a **New registration**. The Redirect URI empty does not need to be specified.



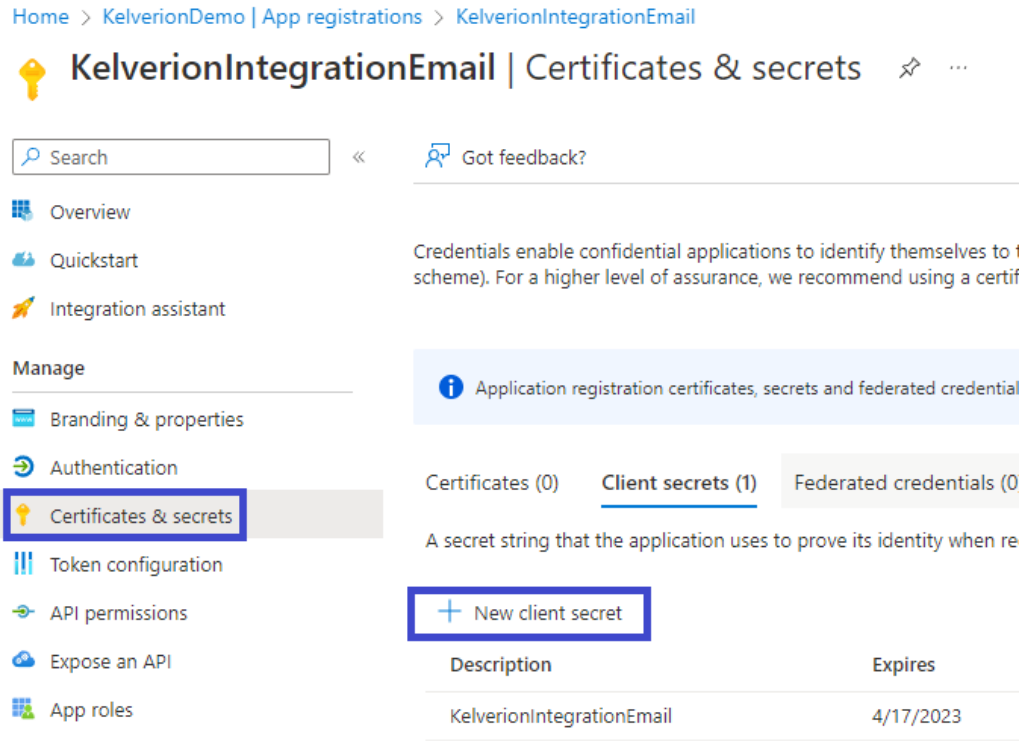
3. After the app registration is created, from the **Overview** tab record the following settings:

- Display name
- Application (client) ID
- Directory (tenant) ID

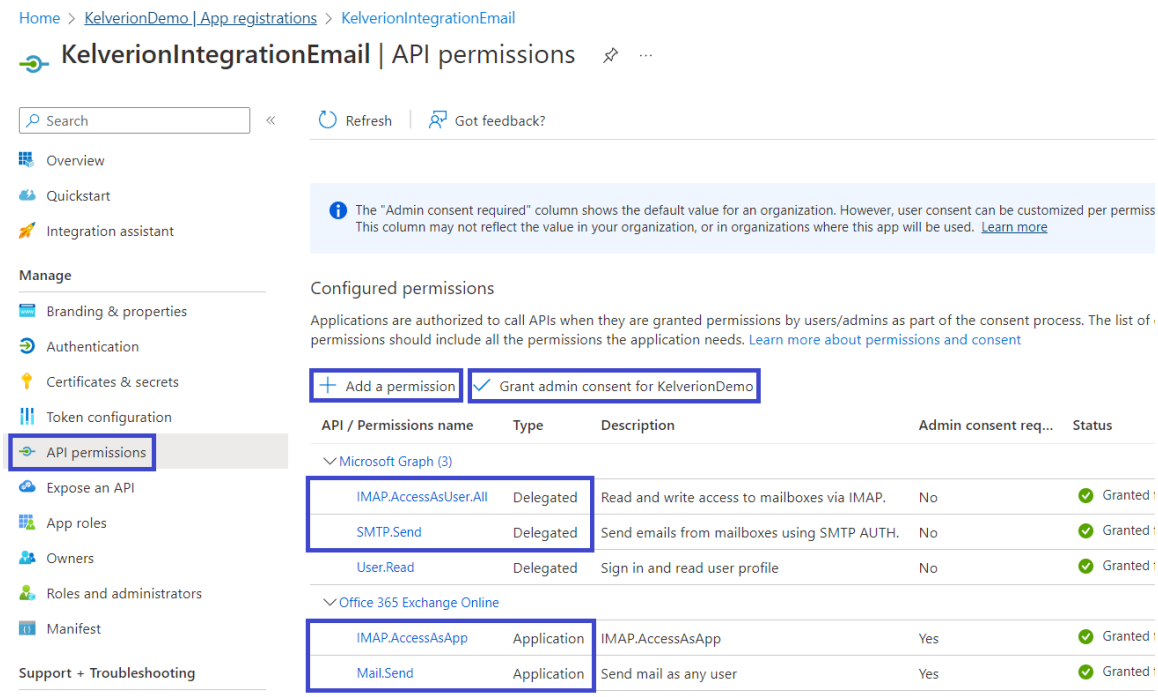
The Directory (tenant) ID and Application (client) ID are needed for setting the **Incoming/Outgoing Tenant ID** and **Incoming/Outgoing Client ID**, respectively, in the Integration Pack connection options. For details, please see Configuring the Kelverion Integration Pack for OAuth Email.

The Display name, Directory (tenant) ID, Application (client) ID are needed when you Grant Exchange Online Permissions.

4. In the **Certificates & secrets** tab, create a **New client secret** for the app. Make sure to record the client secret when created, you will not be able to do so afterwards. The secret value must be set in the **Incoming/Outgoing Client Secret** property, when configuring the Integration Pack connection options. For details, please see Configuring the Kelverion Integration Pack for OAuth Email.

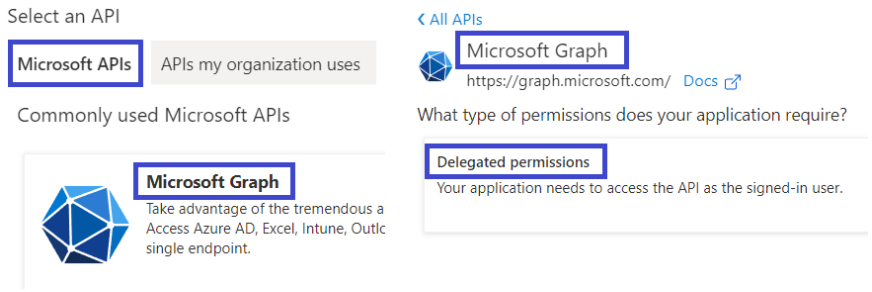


5. In the **Authentication** tab, under **Advanced Settings**, make sure that **Allow public client flows** is enabled.
6. In the **API permissions** tab, configure permissions for the IP to send and read email. Delegated permissions will allow the IP to connect with Username and password. Application permissions will allow the IP to connect with Client ID and Client Secret. After adding the necessary permissions, make sure to **Grant admin consent** for your application.



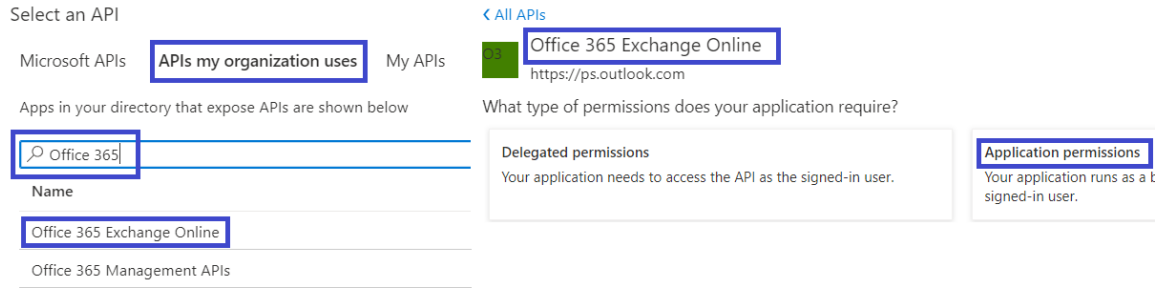
Microsoft Graph <u>Delegated</u> permissions:

- IMAP.AccessAsUser.All
- SMTP.Send



Office 365 Exchange Online <u>Application</u> permissions:

- IMAP.AccessAsApp
- Mail.Send



Grant Exchange Online Permissions

After creating and configuring the Azure AD app, you must register the app service principal with Exchange Online, to grant it permissions to access an email account. To accomplish this, you will need to use the Exchange Online and Azure AD PowerShell modules.

1. Install and import the necessary PowerShell modules:

```
Install-Module -Name ExchangeOnlineManagement -allowprerelease
Install-Module -Name AzureAD
Import-module AzureAD
Import-module ExchangeOnlineManagement
```

2. Connect to Exchange Online and Azure AD. Make sure to use a tenant admin user with sufficient permissions to create service principals and assign permissions in Exchange.

```
$tenantId = "<Your Azure AD Tenant ID>"
Connect-ExchangeOnline -Organization $tenantId
Connect-AzureAD -Tenant $tenantId
```

3. Grant access for your Azure AD app to access the email account used by the Integration Pack:

```
$clientAppName = "Your Azure AD app name"
$servicePrincipalName = "Service principal for <Your Azure AD app>"
$email = "<Email address for account used by Integration Pack>"

$appServicePrincipal = Get-AzureADServicePrincipal
    -SearchString  $clientAppName

New-ServicePrincipal
    -AppId $appServicePrincipal.AppId
    -ServiceId $appServicePrincipal.ObjectId
    -DisplayName $servicePrincipalName

Add-MailboxPermission
    -Identity $email
    -User $appServicePrincipal.ObjectId
    -AccessRights FullAccess
```

4. (Optional) Grant access for your Azure AD app to access another email account used by the Integration Pack:

```
$anotherEmail = "<Another email account used by Integration Pack>"

Add-MailboxPermission
    -Identity $anotherEmail
    -User $appServicePrincipal.ObjectId
    -AccessRights FullAccess
```

Enable SMTP in Exchange Online

By default, the SMTP protocol is disabled in Office 365 Exchange Online. Please follow the steps bellow to enable SMTP for the email account used by the Integration Pack. You will need to use the Exchange Online PowerShell module again. For additional details, please see Microsoft documentation at https://learn.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/authenticated-client-smtp-submission.

1. Connect to Exchange Online using your Office 365 tenant admin user.

```
$tenantId = "<Your Azure AD Tenant ID>"
```

```
Connect-ExchangeOnline -Organization $tenantId
```

2. Enable SMTP for the email account used by the Integration Pack. Repeat this step to enable other email accounts.

```
$email = "<Email address for account to be used by the Integration Pack>"
Set-CASMailbox -Identity $email -SmtpClientAuthenticationDisabled $false
```

Message Identification

The Kelverion Integration Pack for OAuth Email uses Unique Identifiers, to uniquely identify and distinguish email messages. Unique ID values, which are supported by IMAP servers provide a reliable mechanism for managing email messages, however, there are some considerations that users should consider.

Considerations When Using IMAP

When using IMAP to manage email messages, users should be aware that Unique ID values are only valid within a given mailbox (folder) and that an email message will be assigned a new Unique ID value when it is moved to another mailbox. It is also possible that email messages in different mailboxes can be assigned the same Unique ID value. Consequently, runbook designers should take care when building runbooks that move and/or delete email messages to ensure that they are using the correct Unique ID value.

It should also be noted that the Integration Pack for Email incorporates a feature of IMAP, called *UIDVALIDITY,* to help ensure that message UID values are valid from across sessions. Specifically, the message **Unique ID** value published by the integration pack is a combination of the message UID value and the mailbox UIDVALIDITY value. In the rare instance that the integration encounters and invalid UID, the activity will fail with an appropriate error, or in the case of the Monitor Email activity, report a warning. Also in rare occasions, Monitor Email activity will fail to identify new email messages when the retrieved messages have invalid UID values.

Email Activities

This integration pack adds the **KA OAuth Email** category to the **Activities** pane in the Runbook Designer. This category contains the following activities:

- Delete Email
- Empty Mailbox
- Get Email
- Get Email List
- Get Mailbox Info
- Get Mailbox List
- Monitor Email
- Move Email
- Send Email

Common Configuration Instructions for All Activities

The following configuration instructions apply to all activities in this integration pack. Links to this section are included in the configuration instructions for each activity.

Activity Properties

Each activity has a set of required or optional properties that define the configuration of that activity. This includes how it connects to other activity or how the activity performs its actions. You can view or modify activity properties in the Orchestrator Client.:

To configure the properties for an activity:

1. Double-click the activity. Alternatively, you can right-click the activity, and then click **Properties**.
2. To save your configuration entries, click **Finish**.

In the activity properties dialog box, several tabs along the left side provide access to general and specific settings for the activity. Although the number of available tabs for activity properties differs from activity to activity, all activities will have a **General** tab, a **Properties tab and/or Filters** tab, and a **Run Behavior** tab. Some activities may have additional tabs.

General Tab

This tab contains the **Name** and **Description** properties for the activity. By default, the **Name** of the activity is the same as its activity type, and the **Description** is blank. You can modify these properties to create more descriptive names or provide detailed descriptions of the actions of the activity.

Properties/Filters Tab

These tabs contain properties that are specific to the activity.

All activities in this integration pack have the **Configuration Name** property at the top of the **Properties** tab. This property is used to specify the connection to a SQL table.

To configure the Configuration Name proper:

Click the ellipsis **(…)** button next to the **Name** field, and then select the applicable connection name. Connections displayed in the list have been previously configured as described in Configuring the SQL Server Connections.

Filter Behavior

The Monitor and Get activities use filters to determine the values that will invoke a runbook or retrieve activities. Property values of potential candidates are compared to the values of the filters to determine if they meet the criteria. When matching against values, you select one of the available methods of comparison. An option is provided to either match or not match the filter using each method. For example, the "Does not" version of a method causes alerts that do not match the filter to trigger the runbook.

- **Equals**: the property of the object exactly matches the text or number specified in the filter.
- **Does not equal**: the property of the object does not exactly match the text or number specified in the filter.
- **Is less than**: the property of the object is less than the number specified in the filter.
- **Is less than or equal to**: the property of the object is less than or equal to the number specified in the filter.
- **Is greater than**: the property of the object is greater than the number specified in the filter.
- **Is greater than or equal to**: the property of the object is greater than or equal to the number specified in the filter.
- **Contains**: the property of the object contains the exact text specified in the filter. Unlike the Equals behavior, there can be other text surrounding the matching text.
- **Does not contain**: the property of the object does not contain the exact text specified in the filter. Unlike the Equals behavior, there can be other text surrounding the matching text.
- **Starts with**: the property of the object starts with the exact text specified in the filter. Unlike the Equals behavior, there can be other text following the matching text.
- **Ends with**: the property of the object ends with the exact text specified in the filter. Unlike the Equals behavior, there can be other text preceding the matching text.
- **$null is** accepted as a filter value and can be used to build filter conditions which evaluate for properties being equal or not equal to PowerShell $null.

When building DateTime filter conditions it is recommended to use inequality comparison operators. If you use equals or does not equal to compare with DateTime values, ensure the filter value is specified with millisecond precision, otherwise the comparison evaluation may fail.

Run Behavior Tab

This tab contains the properties that determine how the activity manages multi-value published data and what notifications will be sent if the activity fails or runs for an excessive period.

Multi-Value Published Data Behavior

The Get activities retrieve information from another activity or outside source and can return one or more values in the published data. For example, when you use the Get Collection Member activity, the data output from that activity might be a list of computers that belong to the specified collection.

By default, the data from the Get activity will be passed on as multiple individual outputs. This invokes the next activity as many times as there are items in the output. Alternatively, you can provide a single output for the activity by enabling the **Flatten** option. When you enable this option, you also choose a formatting option:

- **Separate with line breaks**. Each item is on a new line. This format is useful for creating human-readable text files for the output.
- **Separate with _** . Each item is separated by one or more characters of your choice.
- **Use CSV format**. All items are in CSV (comma-separated value) format. This format is useful for importing data into spreadsheets or other applications.

The activity will produce a new set of data every time it runs. The **Flatten** feature does not flatten data across multiple instances of the same activity.

Event Notifications

Some activities are expected to take a limited amount of time to complete. If they do not complete within that time they may be stalled or there may be another issue preventing them from completing. You can define the number of seconds to wait for completion of the action. After this period, a platform event will be sent, and the issue will be reported. You can also choose whether to generate a platform event if the activity returns a failure.

To be notified when the activity takes longer than a specified time to run or fails to run:

1. In the **Event Notifications** box, enter the **number of seconds** of run time before a notification is generated.
2. Select **Report if activity fails to run** to generate run failure notifications.

For more information about Orchestrator events, see the “Event Notifications " topics in the Runbook Properties (https://technet.microsoft.com/en-us/library/hh489610.aspx#Event Notifications).

Published Data

Published data is the foundation of a working runbook. It is the data produced because of the actions of an activity. This data is published to an internal data bus that is unique for each runbook. Subsequent activities in the runbook can subscribe to this data and use it in their configuration. Link conditions also use this information to add decision-making capabilities to runbooks.

An activity can subscribe only to data from the activities that are linked before it in the runbook. You can use published data to automatically populate the property values needed by activities.

To use published data:

1. Right-click the property value box, click **Subscribe**, and then click **Published Data**.
2. Click the **Activity** drop-down box and select the activity from which you want to obtain the data.
3. To view additional data elements common to all activities, select **Show Common Published Data**.
4. Click the published data element that you want to use, and then click **OK**.

For a list of the data elements published by each activity, see the Published Data tables in the activity topic. For information about the common published data items, see the Published Data (http://technet.microsoft.com/en-us/library/hh403821.aspx).

Delete Email Activity

The **Delete Email** activity can be used in a runbook to delete one or more email messages. The following tables list the required properties and published data for this activity.

Required Properties

You must configure the following properties:

Mailbox Name	Specifies the name of the mailbox where email messages are to be deleted from. When specifying a folder (mailbox) which is listed in the **Trash Folders** property, in the activity configuration, the activity will permanently remove specified messages from the mailbox.
Unique ID	Identifies one or more email messages to be deleted. Use comma (,) or semicolon (;) to separate multiple message IDs. When a list of IDs is specified, the activity will process multiple email messages in a single session, thus reducing network traffic and improving performance. ***Tip:*** Use the Flatten option under Run Behavior tab of activity properties, to combine multiple Unique IDs published by **Get Email List** activity into a single comma (,) or semi-colon (;) separated list of IDs.
Permanent Remove	Specifies whether messages will be permanently deleted or whether they will be moved to another mailbox. This property is not available when **Mailbox Name** is configured with a folder which is listed in under **Trash Folders** property in the activity configuration.
Move To	Specifies the target mailbox where messages are to be moved. This property is only available when **Permanent Remove** is configured as **False**.

Published Data

The **Delete Email** activity generates the following activity specific published data items:

Unique ID	Unique ID(s) of the messages that were deleted/moved.
Message Count	Number of messages that were deleted/moved.

Empty Mailbox Activity

The **Empty Mailbox** activity can be used in a runbook to permanently remove email messages from a specified mailbox**.** This activity can only be used with IMAP configurations.

Required Properties

You must configure the following properties:

Mailbox Name	Specifies the name of the mailbox to remove messages from.

Published Data

The **Empty Mailbox** activity generates the following activity specific published data items:

Unique ID	Unique ID(s) of the messages that were removed from the mailbox.
Message Count	Number of messages that were removed from the mailbox.

Get Email Activity

The **Get Email** activity can be used in a runbook to download the contents of email messages, including message body and attachments.

Required Properties

You must configure the following properties:

Unique ID	Identifies one or more email messages to be retrieved. Use comma (,) or semicolon (;) to separate multiple message IDs. When a list of IDs is specified, the activity will process multiple email messages in a single session, thus reducing network traffic and improving performance. ***Tip:*** Use the Flatten option under Run Behavior tab of activity properties to combine multiple Unique IDs published by **Get Email List Activity** into a single comma (,) or semi-colon (;) separated list of IDs.
Mailbox Name	Specifies the name of the mailbox that contains the email message(s).

Optional Properties

You can configure the following properties as required:

As Plain Text	Specifies that the plain text message body should be published for an email message flagged as HTML. When a sender does not include a plain text version, **Message Body** will return as empty.
Attachment Download Folder	Specifies a folder where email attachments are to be downloaded. If the folder does not exist, the activity creates it.
If Attachment Exists	Specifies what action to take if a file with the same name already exists in the folder where an attachment is to be saved. • **Rename** – adjusts the name of the attachment so that it does not conflict with existing file name. • **Overwrite** – overwrites existing file with new attachment file. • **Ignore** – does not save the attachment and leaves the exiting file un-changed.

Published Data

The **Get Email** activity generates the following activity specific published data items:

Attachment Count	Number of attachments contained by a message.
Attachment Download Folder	Folder where message attachment(s) were saved.
Bcc	Comma (,) separated list of email addresses specifying hidden recipient(s) of the email message.
Cc	Comma (,) separated list of email addresses specifying other recipients of the email message.

Date Received	Date and time when the email message was received.
Date Received in UTC	UTC date and time when the email message was received.
Date Sent	Date and time when the email message was sent.
Date Sent in UTC	UTC date and time when the email message was sent.
From	Comma (,) separated list of email addresses specifying author(s) of the email message.
HTML	Indicates if the Message Body is in HTML format.
Message Body	Body of the email message.
Message Count	Number of messages retrieved by the activity.
Message ID	Uniquely identifies the email message.
Priority	Message priority.
Saved Attachments	Comma (,) separated list of attachment file names.
Sequence Number	Sequence number of the email message.
Size (Bytes)	The size of the mail message in bytes.
Subject	The subject of the email.
To	Semicolon (;) separated list of email addresses specifying primary recipients of the email message.
Unique Id	Uniquely identifies the email message.

Get Email List Activity

The **Get Email List** activity is used in a runbook to retrieve a list of email messages from a mailbox. This activity retrieves information about email messages; however, it does not download message body or attachments. Typically, you would use Get Email List activity to retrieve and filter email messages and subsequently you would use Get/Delete/Move Email activities to further process those messages.

Tip: Use the Flatten option under Run Behavior tab of activity properties to combine multiple Unique IDs published by Get Email List activity into a single comma (,) or semi-colon (;) separated list of IDs. You can then feed this list into the Unique ID property of activities which accept ID lists, such as Get/Move/Delete Email, so that multiple messages are processed together. Doing so will reduce network traffic and improve performance of the IP.

Required Properties

You must configure the following properties:

Mailbox Name	Specifies the name of the mailbox from which to retrieve email messages.

Published Data

The **Get Email List** activity generates the following activity specific published data items:

Bcc	Comma (,) separated list of email addresses specifying hidden recipient(s) of the email message.
Cc	Comma (,) separated list of email addresses specifying other recipients of the email message.
Date Received	Date and time when the email message was received.
Date Received in UTC	UTC date and time when the email message was received.
Date Sent	Date and time when the email message was sent.
Date Sent in UTC	UTC date and time when the email message was sent.
From	Comma (,) separated list of email addresses specifying author(s) of the email message.
Is Deleted	Indicates if the email message is marked as “deleted”.
Is Draft	Indicates if the email message is a draft.
Is Read	Indicates if the email message has been read.
Message Count	Specifies the number of email messages retrieved by the activity.
Message ID	Uniquely identifies the email message.
Sequence Number	Sequence number of the email message.
Size (Bytes)	Email message size in bytes.
Subject	Email message subject.

To	Comma (,) separated list of email addresses specifying primary recipients of the email message.
Unique Id	Uniquely identifies the email message.

Filters

The **Get Email List** activity provides the following filters that can be used to select which email records to retrieve.

Bcc	Filters email messages by hidden recipient(s). Filtering is performed client-side.
Cc	Filters email messages by other recipient(s). Filtering is performed client-side.
Date Received	Filters email messages by Date Received. Filtering is performed client-side.
Date Received in UTC	Filters email messages by UTC Date Received. Filtering is performed client-side.
Date Sent	Filters email messages by Date Sent. Filtering is performed client-side.
Date Sent in UTC	Filters email messages by UTC Date Sent. Filtering is performed client-side.
From	Filters email messages by author(s). Server-side operations: Equals, Contains, Starts with, Ends with. Note that for these operations, the activity fails to return results when the filter value includes angled brackets characters <>. Client-side operations: Does not equal, Does not contain, Matches pattern, Does not match pattern.
Is Deleted	Filters email messages by Is Deleted flag. Filtering is performed server-side.
Is Draft	Filters email messages by Is Draft flag. Filtering is performed server-side.
Is Read	Filters email messages by Is Read flag. Filtering is performed server-side.
Message ID	Filters email messages by Message ID. Filtering is performed client-side.
Sequence Number	Filters email messages by Sequence Number. Filtering is performed server-side.
Size (Bytes)	Filters email messages by Size. Filtering is performed server-side.
Subject	Filters email messages by Subject. Server-side operations: Equals, Contains, Starts with, Ends with. Client-side operations: Does not equal, Does not contain, Matches pattern, Does not match pattern

To	Filters email messages by primary recipient(s). Filtering is performed client-side.
Unique Id	Filters email messages by Unique ID. Filtering is performed server-side.

Get Mailbox Info Activity

The **Get Mailbox Info** activity can be used in a runbook to retrieve information about a specified mailbox. Typically, you would use this activity to make sure that a mailbox does not grow too large. The performance of the IP can be negatively affected when working with a large mailbox which contains many messages.

Required Properties

You must configure the following properties:

Mailbox Name	Specifies the name of the IMAP mailbox.

Published Data

The **Get Mailbox Info** activity generates the following activity specific published data items:

Mailbox Name	The name of the mailbox.
Message Count	The number of messages present in the mailbox.
Unread Message Count	The number of unread messages in the mailbox.

Get Mailbox List Activity

The **Get Mailbox List** activity can be used in a runbook to retrieve the mailbox names using filter criteria that you specify. This activity is

Required Properties

You must configure the following properties:

Name	Filters mailboxes by mailbox name

Published Data

The **Get Mailbox List** activity generates the following activity specific published data items:

Mailbox Count	The number of mailbox names that were retrieved
Mailbox Name	The name of the mailbox

Monitor Email Activity

The **Monitor Email** activity can be used in a runbook to monitor a mailbox for new email messages. The monitor performance is directly affected by the size of the mailbox it monitors, therefore, to optimize performance, it is recommended to keep the mailbox clean and delete messages when they are no longer needed. You can use the **Get Mailbox Info** activity to check the size and number of messages in a mailbox.

Tip: If you experience connectivity errors when starting the monitor, retry starting the monitor after a few seconds. Once the monitor has started, it will recover automatically and resume monitoring, after encountering server connection issues. If an email is received while the monitor is experiencing server errors, the monitor should detect the message after it recovers. The monitor will log platform evens when encountering server errors.

Required Properties

You must configure the following properties:

Mailbox Name	Specifies the name of the IMAP mailbox to monitor.
Monitor Interval (seconds)	Specifies how often the monitor checks if new emails have arrived on the server.

Published Data

The **Monitor Email** activity generates the following activity specific published data items:

Bcc	Comma (,) separated list of email addresses specifying hidden recipient(s) of the email message.
Cc	Comma (,) separated list of email addresses specifying other recipients of the email message.
Date Received	Date and time when the email message was received.
Date Received in UTC	UTC date and time when the email message was received.
Date Sent	Date and time when the email message was sent.
Date Sent in UTC	UTC date and time when the email message was sent.
From	Comma (,) separated list of email addresses specifying author(s) of the email message.
Is Deleted	Indicates if the email message is marked as “deleted”.
Is Draft	Indicates if the email message is a draft.
Is Read	Indicates if the email message has been read.
Message Count	Specifies the number of email messages retrieved by the monitor.
Message ID	Uniquely identifies the email message.
Sequence Number	Sequence number of the email message.

Size (Bytes)	Email message size in bytes.
Subject	Email message subject.
To	Comma (,) separated list of email addresses specifying primary recipients of the email message.
Unique Id	Uniquely identifies the email message.

Filters

The **Monitor Email** activity provides the following filters that can be used to select which email records will trigger the monitor.

Bcc	Filters email messages by hidden recipient(s). Filtering is performed client-side.
Cc	Filters email messages by other recipient(s). Filtering is performed client-side.
Date Received	Filters email messages by Date Received. Filtering is performed client-side.
Date Received in UTC	Filters email messages by UTC Date Received. Filtering is performed client-side.
Date Sent	Filters email messages by Date Sent. Filtering is performed client-side.
Date Sent in UTC	Filters email messages by UTC Date Sent. Filtering is performed client-side.
From	Filters email messages by author(s). Server-side operations: Equals, Contains, Starts with, Ends with. Note that for these operations, the activity fails to return results when the filter value includes angled brackets characters <>. Client-side operations: Does not equal, Does not contain, Matches pattern, Does not match pattern.
Is Deleted	Filters email messages by Is Deleted flag. Filtering is performed server-side.
Is Draft	Filters email messages by Is Draft flag. Filtering is performed server-side.
Is Read	Filters email messages by Is Read flag. Filtering is performed server-side.
Message ID	Filters email messages by Message ID. Filtering is performed client-side.
Sequence Number	Filters email messages by Sequence Number. Filtering is performed server-side.
Size (Bytes)	Filters email messages by Size. Filtering is performed server-side.
Subject	Filters email messages by Subject. Server-side operations: Equals, Contains, Starts with, Ends with. Client-side operations: Does not equal, Does not contain, Matches pattern, Does not match pattern

To	Filters email messages by primary recipient(s). Filtering is performed client-side.
Unique Id	Filters email messages by Unique ID. Filtering is performed server-side.

Move Email Activity

The **Move Email** activity can be used in a runbook to move one or more email messages between folders. This activity is

Required Properties

You must configure the following properties:

Unique ID	Identifies one or more email messages to be moved. Use comma (,) or semicolon (;) to separate multiple message IDs. When a list of IDs is specified, the activity will process multiple email messages in a single session, thus reducing network traffic and improving performance. ***Tip:*** Use the Flatten option under Run Behavior tab of activity properties to combine multiple Unique IDs published by **Get Email List Activity** into a single comma (,) or semi-colon (;) separated list of IDs.
Mailbox Name	Specifies the name of the mailbox where messages are to be moved from.
Move To	Specifies the name of the mailbox where messages are to be moved.

Published Data

The **Move Email** activity generates the following activity specific published data items:

Unique ID	Unique ID(s) of the messages that were moved.
Message Count	Number of messages that were moved.

Send Email Activity

The **Send Email** activity can be used in a runbook to send an email message.

Required Properties

You must configure the following properties:

Attachment Count	Specifies the number of attachments to be sent with this email message. Specify zero when the message does not contain any attachments. Use this property for specifying email attachments statically.
Attachment 1..10	File path for an email attachment. Only available when **Attachment Count** specifies a number greater than 0.
From	One or more email addresses specifying the author(s) of the email message. Multiple email addresses should be comma (,) or semicolon (**;**) separated.
Message Body	Body of the email message.
Subject	Subject of the email message.
To	One or more email addresses specifying the primary recipient(s) of the email message. Multiple email addresses should be comma (,) or semicolon (**;**) separated.

Optional Properties

You can configure the following properties as required:

Additional Attachments	Semicolon (;) separated list of attachment paths. Use this property for specifying email attachments dynamically. The attachments specified by this property are sent in addition to any attachments specified by Attachment 1 to 10 properties.
Bcc	One or more email addresses specifying recipient(s) that should not be visible to the other recipients of the email message. Multiple email addresses should be comma (,) or semicolon (**;**) separated.
Cc	One or more email addresses specifying other recipients(s) of the email message. Multiple email addresses should be comma (,) or semicolon (**;**) separated.
Copy Sent Message To	Specifies the target mailbox where a copy of the sent message should be saved.
HTML	Specifies whether the Message Body is in HTML format.
Priority	Specifies the priority of the message.

Published Data

The **Send Email** activity does not generate any activity specific published data.