User Guide: Microsoft Exchange Online Functions for IBM Resilient v1.3.0

Table of Contents

History

Date	Version	Note
2/2021	1.3.0	Added attachments to send message
12/2020	1.2.0	Performance improvement when querying a large tenant using the "all" query parameter
9/2020	1.1.0	Numerous performance and functional improvements around querying mailboxes
2/2020	1.0.0	Initial publication

Key Features

Resilient Integration with Exchange Online provides the capability to access and manipulate Microsoft Exchange Online (Office 365 in the cloud) messages from the IBM Resilient Soar Platform. The integration uses Microsoft Graph API to access the data in Office 365. Included in the integrations are the following capabilities:

- Get the user profile of the specified email address in JSON format.
- Get a specified message and and return the results in JSON format.
- Get a specified message in .eml format and write as an incident attachment.
- Move a message to a specified "Well-known" Outlook folder.
- Send an message: from the specified email address to the specified recipients with specified message subject and body text.
- Query messages of a single user, a list of users, or the whole tenant and return a list of messages matching the criteria: message sender, messages from a specific Well-known folder, a time frame for when the message was received, text contained in the message subject or the message body, whether the message has attachments. Results are returned in the Exchange Online Query Message Results data table.
- Delete a single specified message from a specified email address.
- Delete a list of messages that are the results of a message query. The messages deleted are written to the Exchange Online Query Messages data table.

- Create a meeting event in the organizer's Outlook calendar and send a calendar event message to meeting participants inviting them to the meeting.

Integration Flow for Phishing Investigation Use Case

The Exchange Online integration primary use case is to monitor and control email activities in Exchange Online (Office 365 Outlook in the cloud) and protect against inbound malicious emails.

To use the integration, run the Query Messages rule from the Action menu of an incident. From the rule, you can search a single email address, a list of email addresses or the entire tenant. Results of the query are returned in the Exchange Online Message Query Results data table on the Exchange Online incident tab. Each row in the data table contains information from one message and the following actions can be performed on each message when it's state is Active in the Status column:

- Create artifacts: Email Recipient, Email Sender, Email Subject.
- Delete the message.
- Move the message to a Well-known folder.
- Write the message .eml as an incident.
- Write the message JSON returned from MS Graph to an incident note.

The data table Status column is set to Active when the message is entered in the table. Any time after that, a user can delete the message; however, this could update the Status field to Not Found or Deleted if the message is deleted when running one of the above data table rules or workflows.

The first column of the data table displays the time the query occurred. You can use this value to sort through data if multiple queries are run and entered into the data table. You may want to empty the data table after each query.

Because a large number of messages can be returned from a query, the integration has following parameters in the app.config to limit the number of messages returned:

- max_messages
- max_users

Considering using these parameters to improve performance when running queries.

You can perform additional investigation, including using other email analysis scripts and integrations, on messages after writing the message to a note or attachment.

Once you complete the investigation of the messages and there are problematic messages that you want to delete, use the Example: Exchange Online Delete Messages for Query Results rule from the incident's Actions menu. The rule starts a workflow that performs a query of messages and sends the matching results to a function that deletes a list of messages. The results are written to the Exchange Online Message Query Results data table with a Status column of "Deleted" in red. An incident note is also written that indicates the number of messages deleted.

Use this rule with caution as you can delete many user messages.

At anytime the user can send a message or schedule a meeting using the Exchange Online: Send Message and Exchange Online: Create meeting rules and workflows.

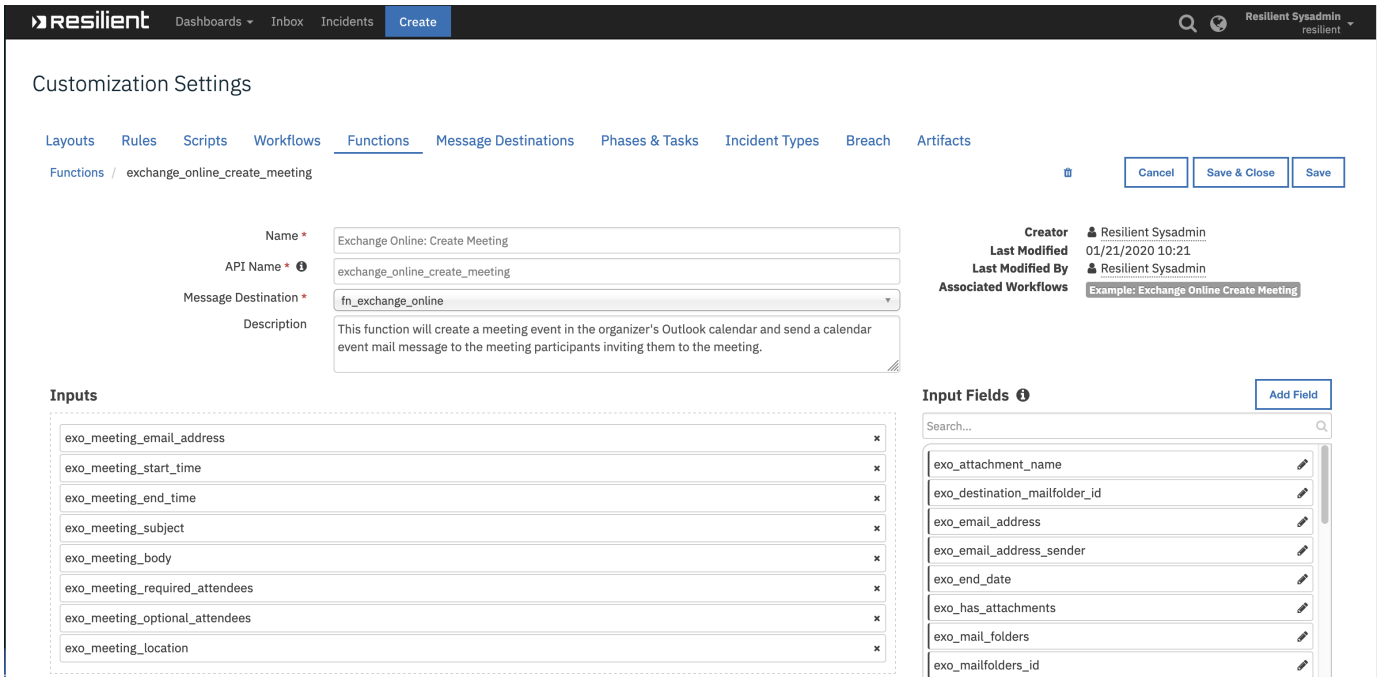
Function - Exchange Online: Create Meeting

Exchange Online: Create Meeting function requires the following Microsoft Graph API Application permissions:

- **Calendars.ReadWrite**
- **MailboxSettings.Read**

The Exchange Online: Create Meeting function creates a meeting event in the organizer's Outlook calendar and sends a calendar event invitation message to the meeting participants.

The meeting start and end times are set in the time zone of the meeting organizer's Outlook mailbox preferred time zone setting. If no mailbox time zone setting is set, the meeting time is calculated using UTC time zone.



▶ Inputs:

Name	Type	Required	Example	Tooltip
exo_meeting_body	text	Yes	meeting message body	Meeting message body
exo_meeting_email_address	text	Yes	user@example.com	Email address of meeting coordinator
exo_meeting_end_time	datetimepicker	Yes	–	End date and time for meeting
exo_meeting_location	text	No	–	-
exo_meeting_optional_attendees	text	No	user1@example.com, user2@example.com	Comma separated list of optional attendee email addresses
exo_meeting_required_attendees	text	No	user1@example.com, user2@example.com	Comma separated list of required attendee email addresses
exo_meeting_start_time	datetimepicker	Yes	–	Meeting start date and time
exo_meeting_subject	text	Yes	–	Meeting Subject

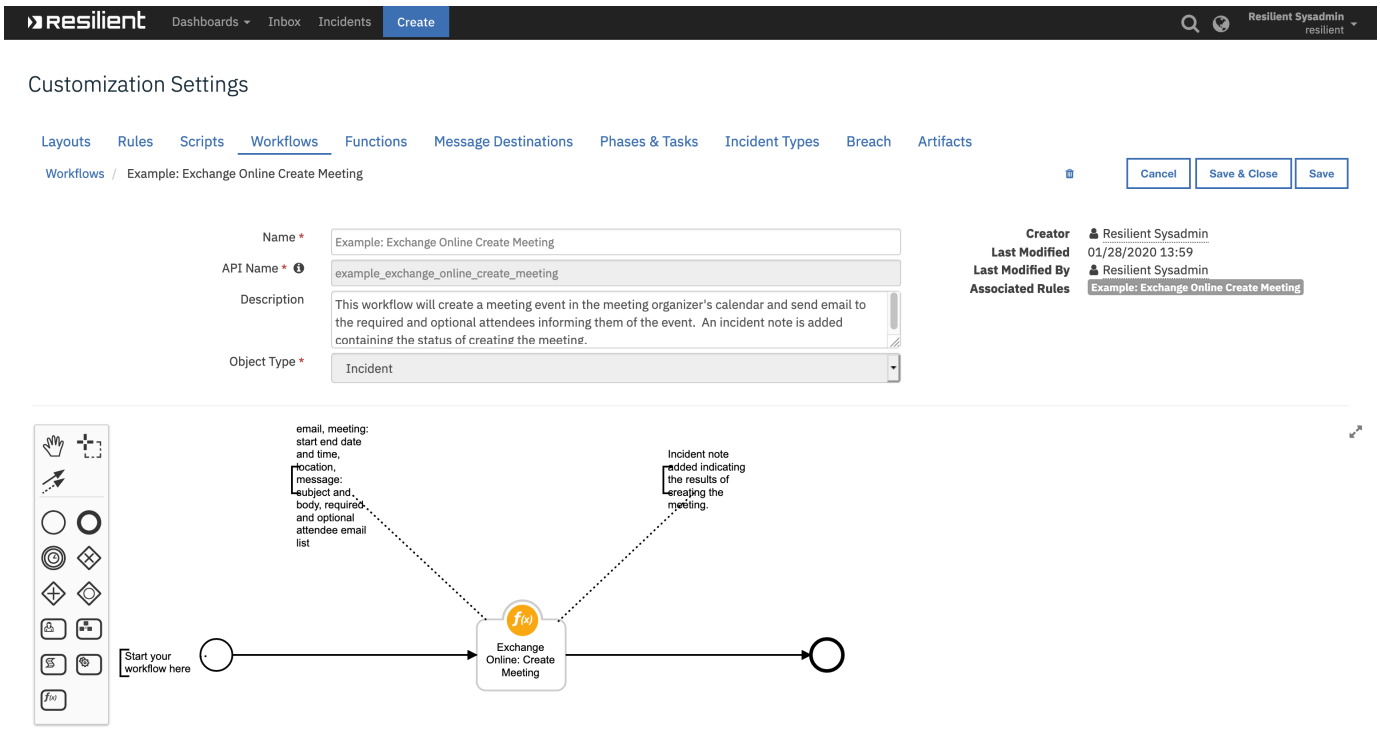
▶ Outputs:

```
results = {
    'inputs': {
          u'exo_meeting_end_time': 1581022800000,
          u'exo_meeting_optional_attendees': None,
          u'exo_meeting_subject': u'phishing meeting',
```

```
          u'exo_meeting_body': u'<div class="rte"><div>We need to talk about this!</div>
</div>', u'exo_meeting_required_attendees':
u'resilient3@securitypocdemos.onmicrosoft.com,
resilient2@securitypocdemos.onmicrosoft.com',
          u'exo_meeting_start_time': 1581004800000,
          u'exo_meeting_email_address': u'resilient2@securitypocdemos.onmicrosoft.com',
u'exo_meeting_location': None},
    'metrics': {'package': 'fn-exchange-online',
          'timestamp': '2020-02-04 13:30:45',
          'package_version': '1.0.0',
          'host': 'MacBook-Pro.local',
          'version': '1.0',
          'execution_time_ms': 1728},
    'success': True,
      'content': {u'body': {u'content': u'', u'contentType': u'html'}, u'sensitivity':
u'normal', u'locations': [], .....}',
    'reason': None,
    'version': '1.0',
    'pretty_string': u'{\n    "body": {\n        "content": "",\n        "contentType":
"html"\n    },\n    "sensitivity": "normal",\n    "locations":
[]"040000008200E00074C5B7101A82E0080000000096E4493689DBD501000000000000000010000000ADDA5C
7497FB60469CE3850456D898C5",\n    "seriesMasterId": null,\n    "responseStatus": {\n
"response": "organizer",\n        "time": "0001-01-01T00:00:00Z"\n    },\n
"@odata.etag": "W/\\"SX/wDQMKnESReIRb/seOFAAAJWnN6Q==\\""\n}',
    'content': {u'body': {u'content': u'', u'contentType': u'html'}, u'sensitivity':
u'normal', u'locations': [],..... "}'
}
```


▶ Workflows:

The Example: Exchange Online Create Meeting workflow calls the create meeting function and then write the results to an incident note.



▶ Example Pre-Process Script:

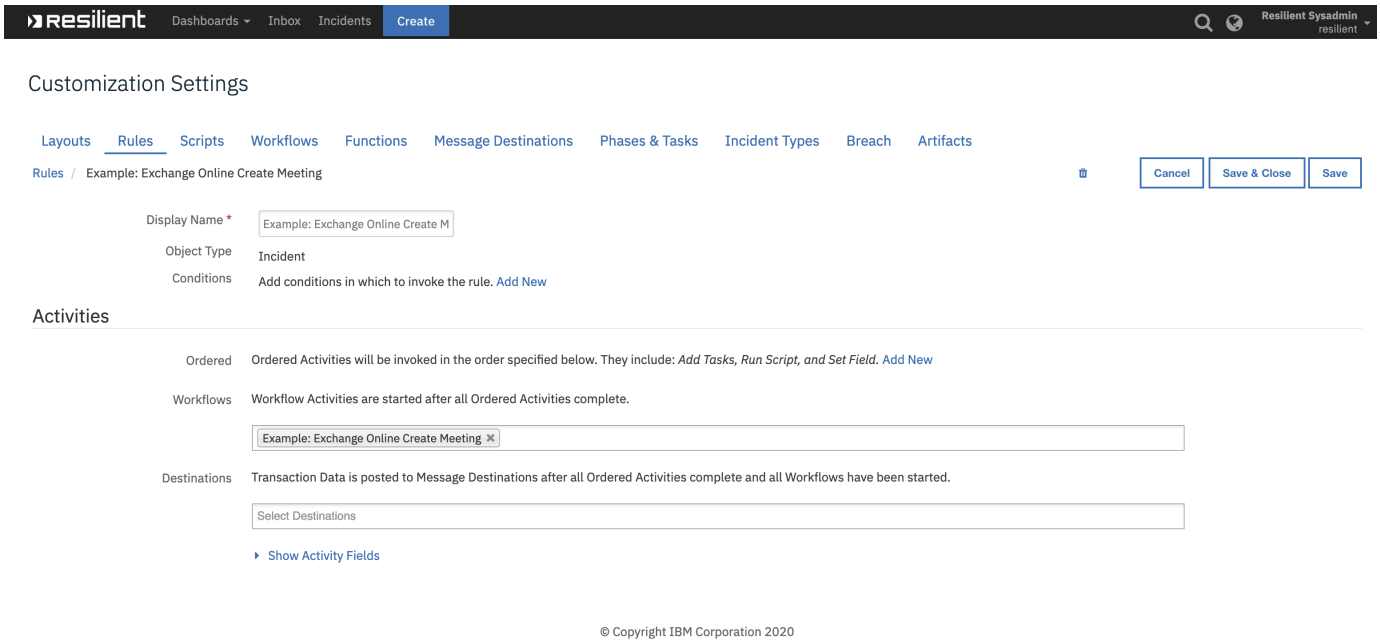
```
inputs.exo_meeting_email_address = inputs.exo_meeting_email_address  if
rule.properties.exo_meeting_email_address is None else
```

```
rule.properties.exo_meeting_email_address
inputs.exo_meeting_start_time = inputs.exo_meeting_start_time if
rule.properties.exo_meeting_start_time is None else
rule.properties.exo_meeting_start_time
inputs.exo_meeting_end_time = inputs.exo_meeting_end_time if
rule.properties.exo_meeting_end_time is None else rule.properties.exo_meeting_end_time
inputs.exo_meeting_subject = inputs.exo_meeting_subject if
rule.properties.exo_meeting_subject is None else rule.properties.exo_meeting_subject
inputs.exo_meeting_body = inputs.exo_meeting_body if
rule.properties.exo_meeting_body.content is None else
rule.properties.exo_meeting_body.content
inputs.exo_meeting_required_attendees = inputs.exo_meeting_required_attendees if
rule.properties.exo_meeting_required_attendees is None else
rule.properties.exo_meeting_required_attendees
inputs.exo_meeting_optional_attendees = inputs.exo_meeting_optional_attendees if
rule.properties.exo_meeting_optional_attendees is None else
rule.properties.exo_meeting_optional_attendees
inputs.exo_meeting_location = inputs.exo_meeting_location if
rule.properties.exo_meeting_location is None else rule.properties.exo_meeting_location
```

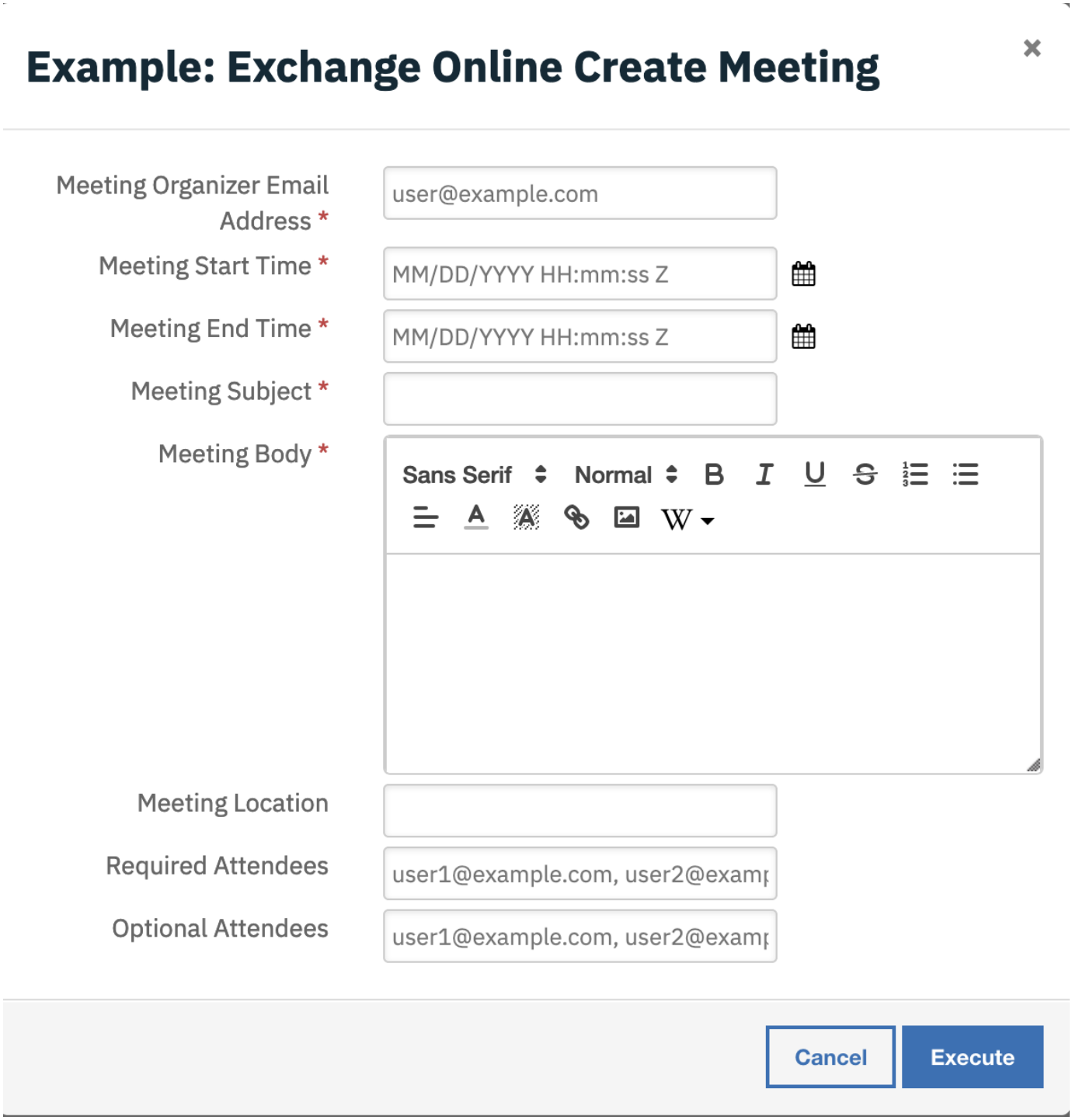
▶ Example Post-Process Script:

```
if results.success:
  noteText = u"Exchange Online created meeting\n   From:
{0}\n{1}".format(results.inputs["exo_meeting_email_address"],results.pretty_string)
else:
  noteText = u"Exchange Online meeting was NOT created\n   From:
{0}\n{1}".format(results.inputs["exo_meeting_email_address"], results.pretty_string)

incident.addNote(noteText)
```

▶ Example Rule:



When the Example: Exchange Online Create Meeting rule is activated the following rule activity popup dialog appears prompting for input for creating the meeting and sending a message to the invitees:

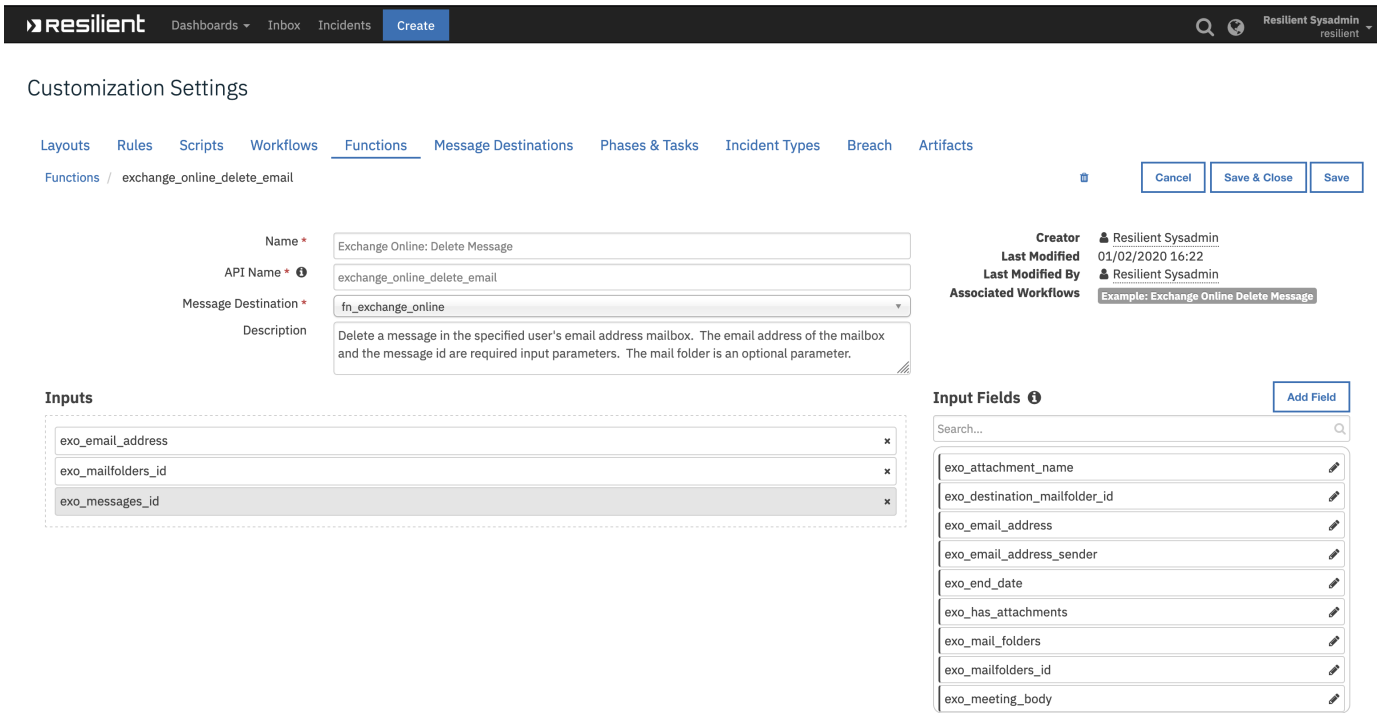


Function - Exchange Online: Delete Message

Exchange Online: Delete Message function requires the following Microsoft Graph API Application permission:

- **Mail.ReadWrite**

Delete a message in the specified user's email address mailbox. The email address of the mailbox and the message ID are required input parameters. The mail folder is an optional parameter.



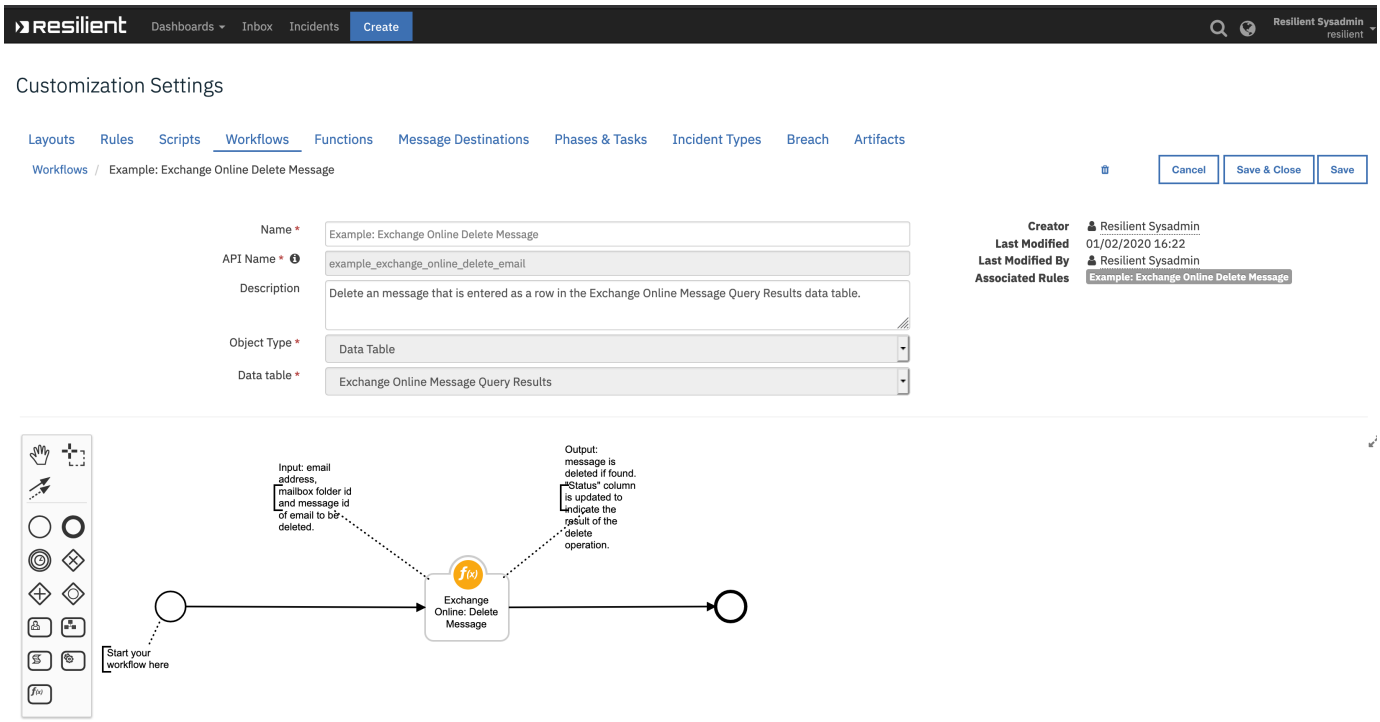
▶ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_mailfolders_id	text	No	inbox	MailFolders ID
exo_messages_id	text	Yes	–	The message ID of the message to be deleted

▶ Outputs:

```
results = {
   'inputs': {u'exo_mailfolders_id': None,
              u'exo_messages_id':
u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
Jf-ANAwqcRJF4hFv_x44UAAAinByvAABJf-ANAwqcRJF4hFv_x44UAAAinIROAAA=', u'exo_email_address':
u'resilient2@securitypocdemos.onmicrosoft.com'},
    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-02-04 09:08:13',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 1300},
    'success': True,
    'content': {'value': True},
    'raw': '{"value": true}',
    'reason': None,
    'version': '1.0'
}
```

▶ Workflows:



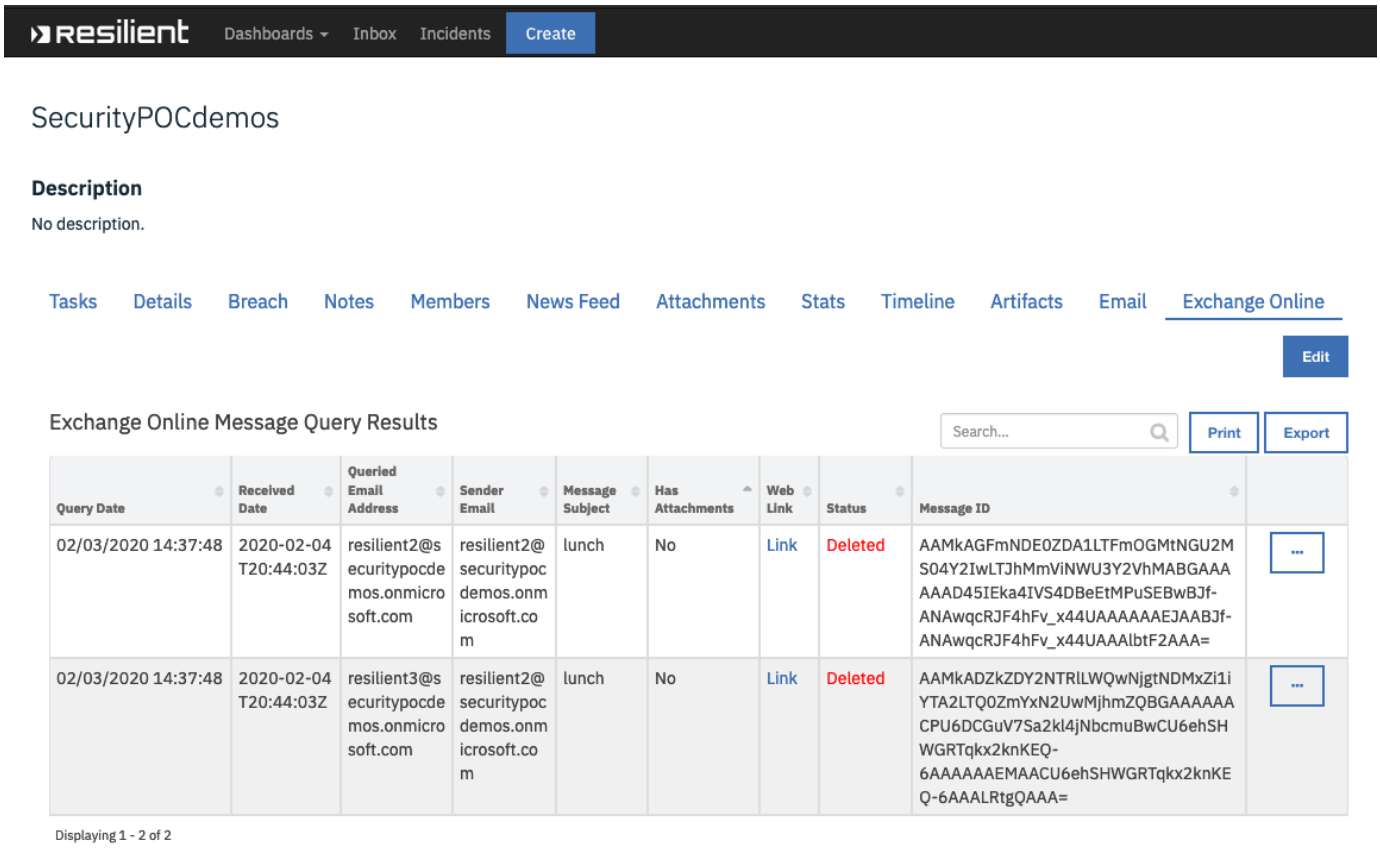
▶ Example Pre-Process Script:

```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
inputs.exo_mailfolders_id = None
```

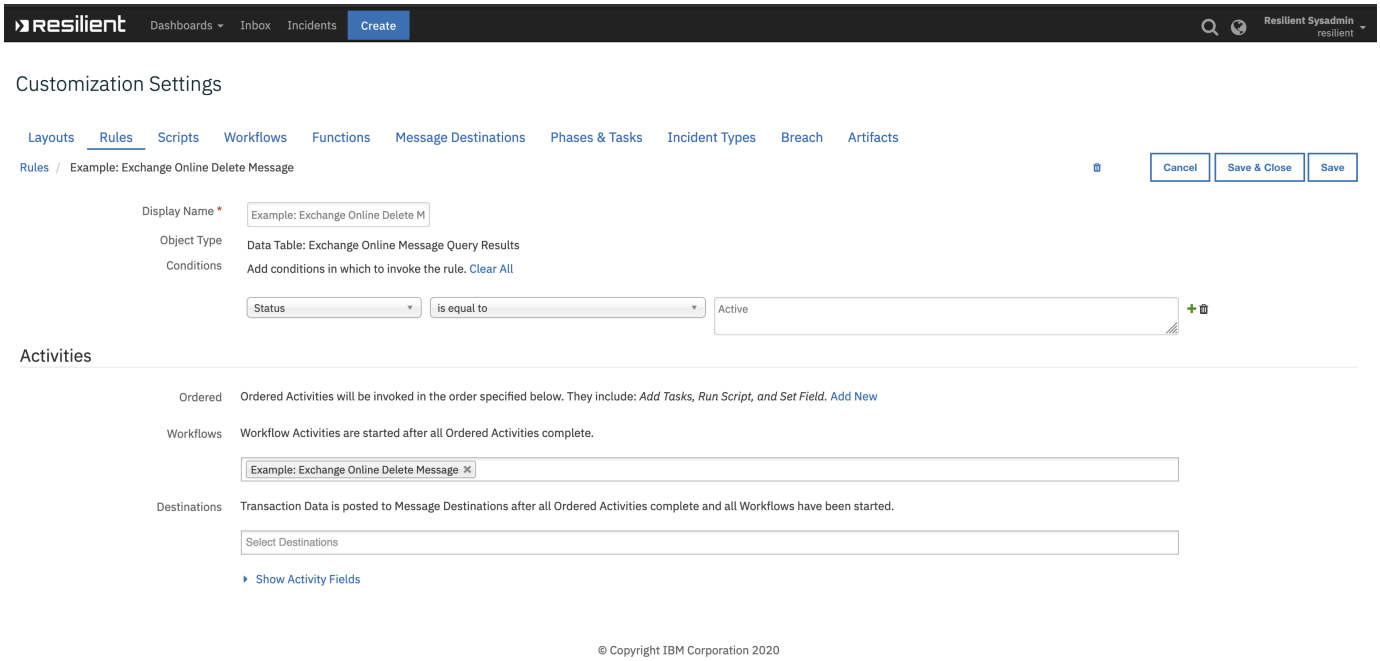
▶ Example Post-Process Script:

```
if results.success:
  # The message was deleted, so update "status" column in data table.
  text = u"""<p style= "color:{color}">{status} </p>""".format(color="red",
status="Deleted")
  row['exo_dt_status'] = helper.createRichText(text)
  row['exo_dt_web_link'] = ""
elif results.content["error"] is not None:
  # There is an "item not found" error mostly likely here
  row['exo_dt_status'] = helper.createRichText(results.content["error"]["code"])
  row['exo_dt_web_link'] = ""
```

▶ Example Workflow Output:



▶ Example Rule:

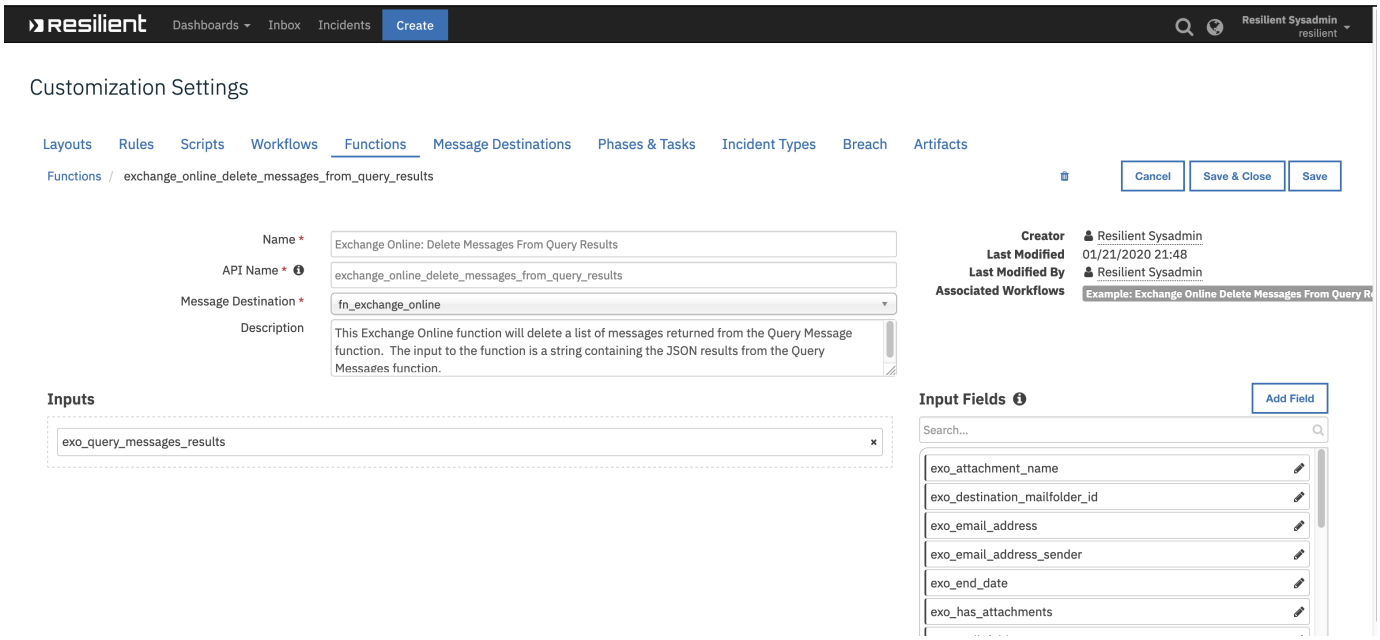


Function - Exchange Online: Delete Messages From Query Results

Exchange Online: Delete Messages From Query Results function requires the following Microsoft Graph API Application permissions:

- **Mail.ReadWrite**
- **User.Read.All** (if querying the whole tenant)

This Exchange Online function deletes a list of messages returned from the Query Message function. The input to the function is a string containing the JSON results from the Query Messages function.



▶ Inputs:

Name	Type	Required	Example	Tooltip
exo_query_messages_results	text	Yes	–	String containing JSON data results from Query Messages function

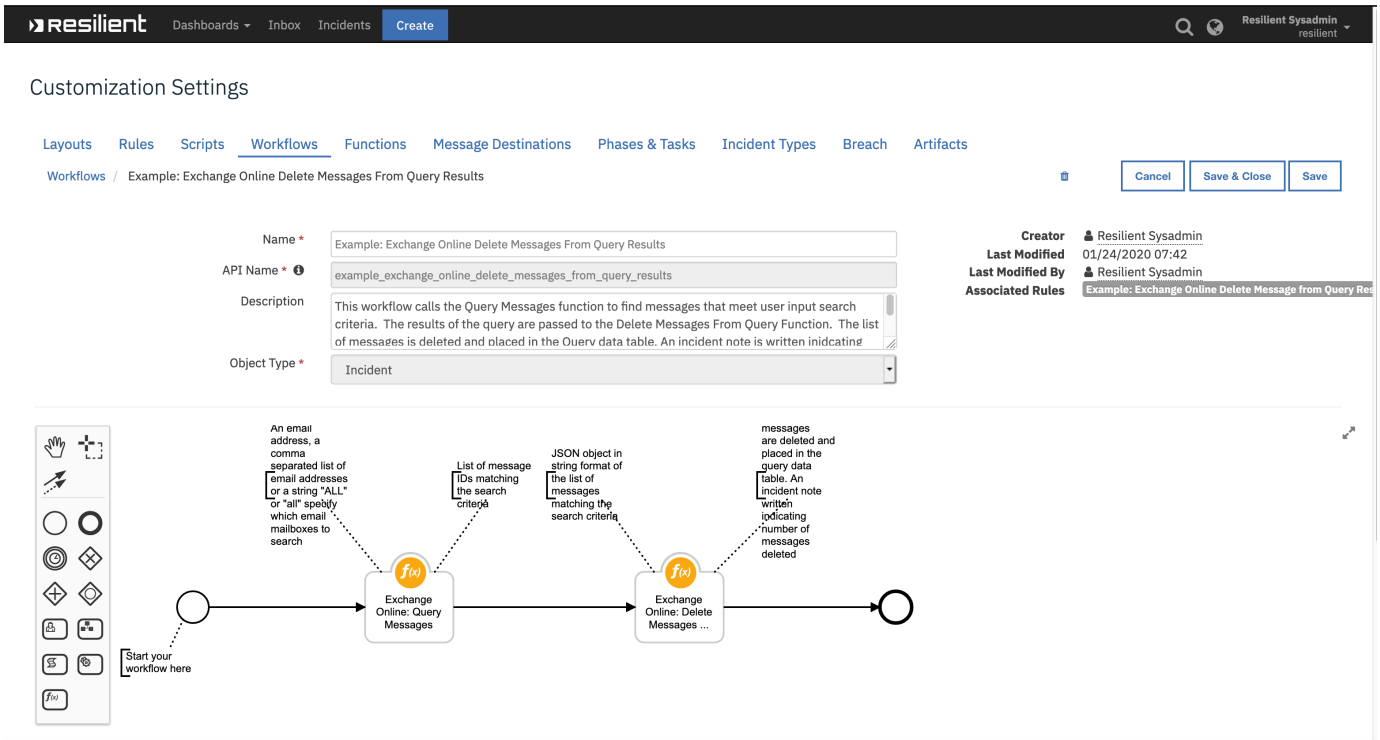
▶ Outputs:


```
results = {'inputs': {u'exo_query_messages_results': u'[{"status_code": 200,
"email_address": "resilient3@securitypocdemos.onmicrosoft.com", "email_list":
[{"sentDateTime": "2020-02-05T20:03:21Z", "conversationId":
"AAQkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAQAExiSXKQgttDm_-WozAl5As=",
"internetMessageId": "
<MWHPR2201MB11357FA62DE5F7C1B1EBCF53B1020@MWHPR2201MB1135.namprd22.prod.outlook.com>",
"id":
"AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbcmuBwCU
6ehSHWGRTqkx2knKEQ-6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ-6AAALRtgSAAA=",
"isReadReceiptRequested": false, "subject": "lunch", "lastModifiedDateTime": "2020-02-
05T20:05:13Z", "bodyPreview": "Do you want to meet for lunch?", "from": {"emailAddress":
{"name": "Resilient User 2", "address": "resilient2@securitypocdemos.onmicrosoft.com"}},
"flag": {"flagStatus": "notFlagged"}, "isDraft": false, "replyTo": [], "changeKey":
"CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT", "receivedDateTime": "2020-02-05T20:03:24Z",
"parentFolderId":
"AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAuAAAAAACPU6DCGuV7Sa2kl4jNbcmuAQCU
6ehSHWGRTqkx2knKEQ-6AAAAAAEMAAA=", "body": {"content": "<html>\\r\\n<head>\\r\\n<meta
http-equiv=\\"Content-Type\\" content=\\"text/html; charset=utf-8\\">\\r\\n<meta
content=\\"text/html; charset=iso-8859-1\\">\\r\\n<style type=\\"text/css\\"
style=\\"display:none\\">\\r\\n<!--\\r\\np\\r\\n\\t{margin-top:0;\\r\\n\\tmargin-
bottom:0}\\r\\n-->\\r\\n</style>\\r\\n</head>\\r\\n<body dir=\\"ltr\\">\\r\\n<div
style=\\"font-family:Calibri,Arial,Helvetica,sans-serif; font-size:12pt;
color:rgb(0,0,0)\\">\\r\\nDo you want to meet for lunch?
<br>\\r\\n</div>\\r\\n</body>\\r\\n</html>\\r\\n", "contentType": "html"},
"isDeliveryReceiptRequested": false, "importance": "normal", "toRecipients":
[{"emailAddress": {"name": "Resilient User 3", "address":
"resilient3@securitypocdemos.onmicrosoft.com"}}], "ccRecipients": [], "isRead": false,
"categories": [], "sender": {"emailAddress": {"name": "Resilient User 2", "address":
"resilient2@securitypocdemos.onmicrosoft.com"}}, "createdDateTime": "2020-02-
05T20:03:23Z", "webLink": "https://outlook.office365.com/owa/?
ItemID=AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbc
muBwCU6ehSHWGRTqkx2knKEQ%2F6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ%2F6AAALRtgSAAA%3D&exvsurl=1&vi
ewmodel=ReadMessageItem", "conversationIndex": "AQHV3F9QTGJJcpCC20Ob79ajMCXkCw==",
```

```
"hasAttachments": false, "bccRecipients": [], "inferenceClassification": "focused",
"@odata.etag": "W/\\"CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT\\""}]}]'},

'metrics': {'package': 'fn-exchange-online',
            'timestamp': '2020-02-05 15:06:25',
            'package_version': '1.0.0',
            'host': 'MacBook-Pro.local',
            'version': '1.0',
            'execution_time_ms': 4869},

'success': True,
'content': [{'not_deleted_list': [], 'deleted_list': [{u'sentDateTime': u'2020-02-
05T20:03:21Z', u'webLink': u'https://outlook.office365.com/owa/?
ItemID=AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbc
muBwCU6ehSHWGRTqkx2knKEQ%2F6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ%2F6AAALRtgSAAA%3D&exvsurl=1&vi
ewmodel=ReadMessageItem', u'conversationId':
u'AAQkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAQAExiSXKQgttDm_-WozAl5As=',
u'internetMessageId':
u'<MWHPR2201MB11357FA62DE5F7C1B1EBCF53B1020@MWHPR2201MB1135.namprd22.prod.outlook.com>',
u'id':
u'AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbcmuBwC
U6ehSHWGRTqkx2knKEQ-6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ-6AAALRtgSAAA=',
u'isReadReceiptRequested': False, u'subject': u'lunch', u'lastModifiedDateTime': u'2020-
02-05T20:05:13Z', u'bodyPreview': u'Do you want to meet for lunch?', u'from':
{u'emailAddress': {u'name': u'Resilient User 2', u'address':
u'resilient2@securitypocdemos.onmicrosoft.com'}}, u'isDraft': False, u'importance':
u'normal', u'changeKey': u'CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT',
u'receivedDateTime': u'2020-02-05T20:03:24Z', u'parentFolderId':
u'AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAuAAAAAACPU6DCGuV7Sa2kl4jNbcmuAQC
U6ehSHWGRTqkx2knKEQ-6AAAAAAEMAAA=', u'body': {u'content': u'<html>\r\n<head>\r\n<meta
http-equiv="Content-Type" content="text/html; charset=utf-8">\r\n<meta
content="text/html; charset=iso-8859-1">\r\n<style type="text/css"
style="display:none">\r\n<!--\r\np\r\n\t{margin-top:0;\r\n\tmargin-bottom:0}\r\n--
>\r\n</style>\r\n</head>\r\n<body dir="ltr">\r\n<div style="font-
family:Calibri,Arial,Helvetica,sans-serif; font-size:12pt; color:rgb(0,0,0)">\r\nDo you
want to meet for lunch?<br>\r\n</div>\r\n</body>\r\n</html>\r\n', u'contentType':
u'html'}, u'isDeliveryReceiptRequested': False, u'replyTo': [], u'toRecipients':
[{u'emailAddress': {u'name': u'Resilient User 3', u'address':
u'resilient3@securitypocdemos.onmicrosoft.com'}}], u'ccRecipients': [], u'flag':
{u'flagStatus': u'notFlagged'}, u'categories': [], u'sender': {u'emailAddress': {u'name':
u'Resilient User 2', u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
u'createdDateTime': u'2020-02-05T20:03:23Z', u'isRead': False, u'conversationIndex':
u'AQHV3F9QTGJJcpCC20Ob79ajMCXkCw==', u'hasAttachments': False, u'bccRecipients': [],
u'inferenceClassification': u'focused', u'@odata.etag':
u'W/"CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT"'}], 'email_address':
u'resilient3@securitypocdemos.onmicrosoft.com'}], 'raw': '[{"not_deleted_list": [],
"deleted_list": [{"sentDateTime": "2020-02-05T20:03:21Z", "webLink":
"https://outlook.office365.com/owa/?
ItemID=AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbc
muBwCU6ehSHWGRTqkx2knKEQ%2F6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ%2F6AAALRtgSAAA%3D&exvsurl=1&vi
ewmodel=ReadMessageItem", "conversationId":
"AAQkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAQAExiSXKQgttDm_-WozAl5As=",
"internetMessageId": "
<MWHPR2201MB11357FA62DE5F7C1B1EBCF53B1020@MWHPR2201MB1135.namprd22.prod.outlook.com>",
"id":
"AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQBGAAAAAACPU6DCGuV7Sa2kl4jNbcmuBwCU
6ehSHWGRTqkx2knKEQ-6AAAAAAEMAACU6ehSHWGRTqkx2knKEQ-6AAALRtgSAAA=",
"isReadReceiptRequested": false, "subject": "lunch", "lastModifiedDateTime": "2020-02-
05T20:05:13Z", "bodyPreview": "Do you want to meet for lunch?", "from": {"emailAddress":
{"name": "Resilient User 2", "address": "resilient2@securitypocdemos.onmicrosoft.com"}},
"isDraft": false, "importance": "normal", "changeKey":
"CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT", "receivedDateTime": "2020-02-05T20:03:24Z",
"parentFolderId":
```

```
"AAMkADZkZDY2NTRlLWQwNjgtNDMxZi1iYTA2LTQ0ZmYxN2UwMjhmZQAuAAAAAACPU6DCGuV7Sa2kl4jNbcmuAQCU
6ehSHWGRTqkx2knKEQ-6AAAAAAEMAAA=", "body": {"content": "<html>\\r\\n<head>\\r\\n<meta
http-equiv=\\"Content-Type\\" content=\\"text/html; charset=utf-8\\">\\r\\n<meta
content=\\"text/html; charset=iso-8859-1\\">\\r\\n<style type=\\"text/css\\"
style=\\"display:none\\">\\r\\n<!--\\r\\np\\r\\n\\t{margin-top:0;\\r\\n\\tmargin-
bottom:0}\\r\\n-->\\r\\n</style>\\r\\n</head>\\r\\n<body dir=\\"ltr\\">\\r\\n<div
style=\\"font-family:Calibri,Arial,Helvetica,sans-serif; font-size:12pt;
color:rgb(0,0,0)\\">\\r\\nDo you want to meet for lunch?
<br>\\r\\n</div>\\r\\n</body>\\r\\n</html>\\r\\n", "contentType": "html"},
"isDeliveryReceiptRequested": false, "replyTo": [], "toRecipients": [{"emailAddress":
{"name": "Resilient User 3", "address": "resilient3@securitypocdemos.onmicrosoft.com"}}],
"ccRecipients": [], "flag": {"flagStatus": "notFlagged"}, "categories": [], "sender":
{"emailAddress": {"name": "Resilient USer 2", "address":
"resilient2@securitypocdemos.onmicrosoft.com"}}, "createdDateTime": "2020-02-
05T20:03:23Z", "isRead": false, "conversationIndex": "AQHV3F9QTGJJcpCC20Ob79ajMCXkCw==",
"hasAttachments": false, "bccRecipients": [], "inferenceClassification": "focused",
"@odata.etag": "W/\\"CQAAABYAAACU6ehSHWGRTqkx2knKEQ/6AAALRVqT\\""}], "email_address":
"resilient3@securitypocdemos.onmicrosoft.com"}]',

'reason': None,
'version': '1.0'
}
```


▶ Workflows:



▶ Example Pre-Process Script:

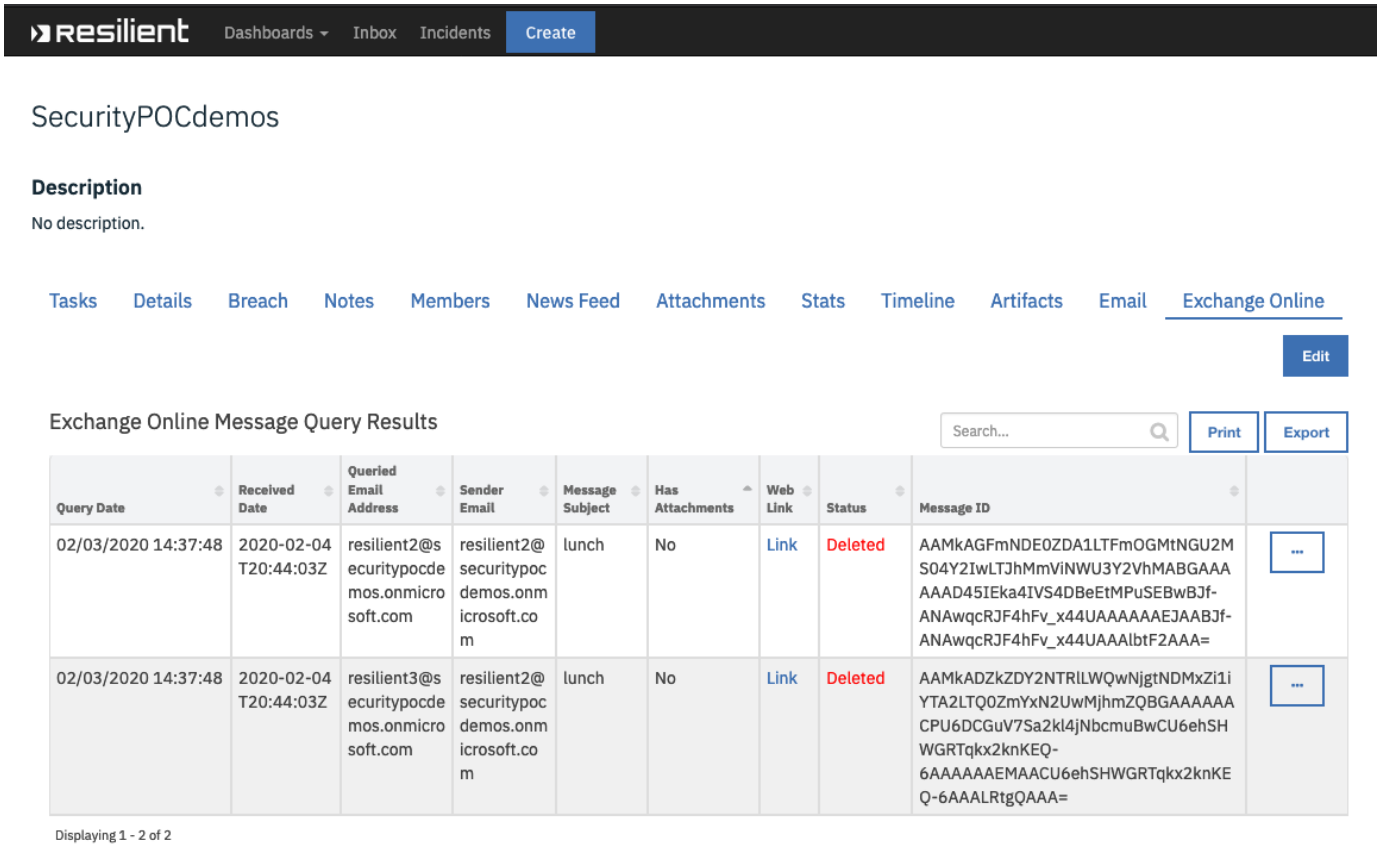
```
inputs.exo_query_messages_results = workflow.properties.exo_query_results['raw']
```

▶ Example Post-Process Script:

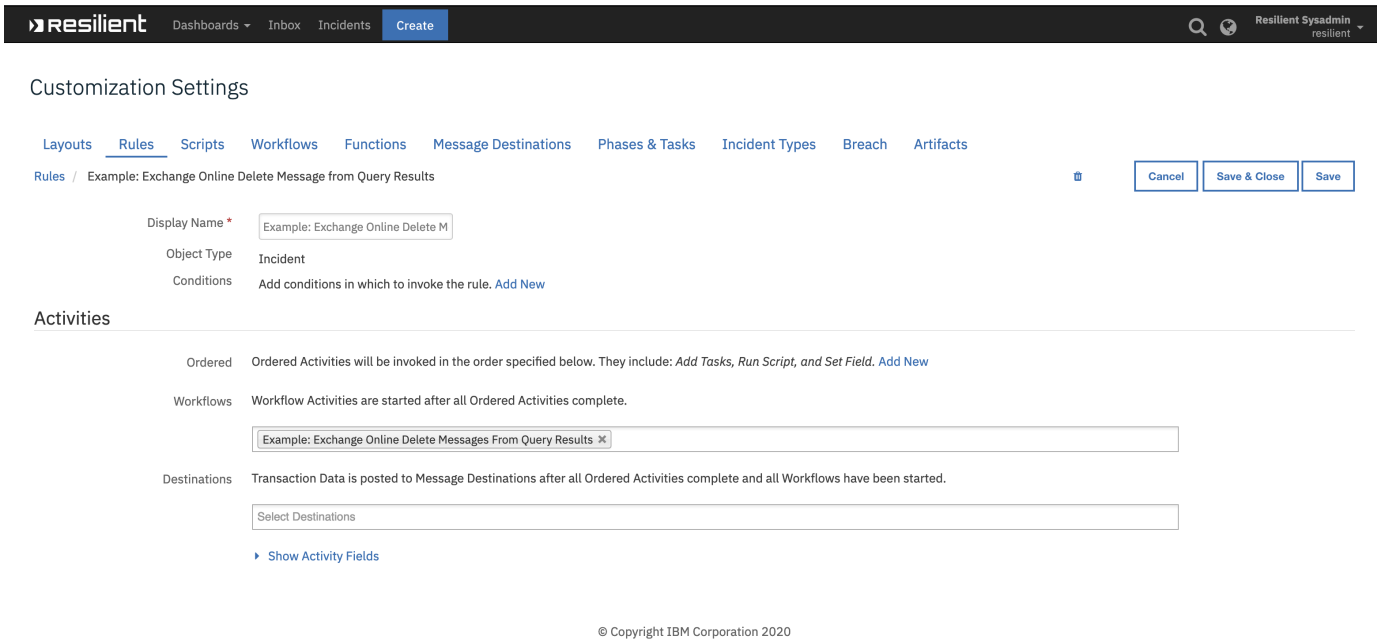
```
from java.util import Date

content = results.get("content")
output_format = content.get("exo_query_output_format")
```

```
# Write to the data table if the user requested it.
if "Exchange Online data table" in output_format:

  user_list = content.get("delete_results")
  # Add each email as a row in the query results data table
  for user in user_list:

    for email in user.get("deleted_list"):
      message_row = incident.addRow("exo_message_query_results_dt")
      message_row.exo_dt_query_date = Date()
      message_row.exo_dt_message_id = email.id
      message_row.exo_dt_received_date   = email.receivedDateTime
      message_row.exo_dt_email_address = user.get("email_address")
      if email.sender:
        message_row.exo_dt_sender_email = email.sender.emailAddress.address
      else:
        message_row.exo_dt_sender_email = ""
      message_row.exo_dt_message_subject = email.subject
      message_row.exo_dt_has_attachments = email.hasAttachments
      if email.webLink:
        ref_html = u"""<a href='{0}'>Link</a>""".format(email.webLink)
        message_row.exo_dt_web_link = helper.createRichText(ref_html)
      else:
        message_row.exo_dt_web_link = ""

      text = u"""<p style= "color:{color}">{status} </p>""".format(color="red",
status="Deleted")
      message_row.exo_dt_status = helper.createRichText(text)
```

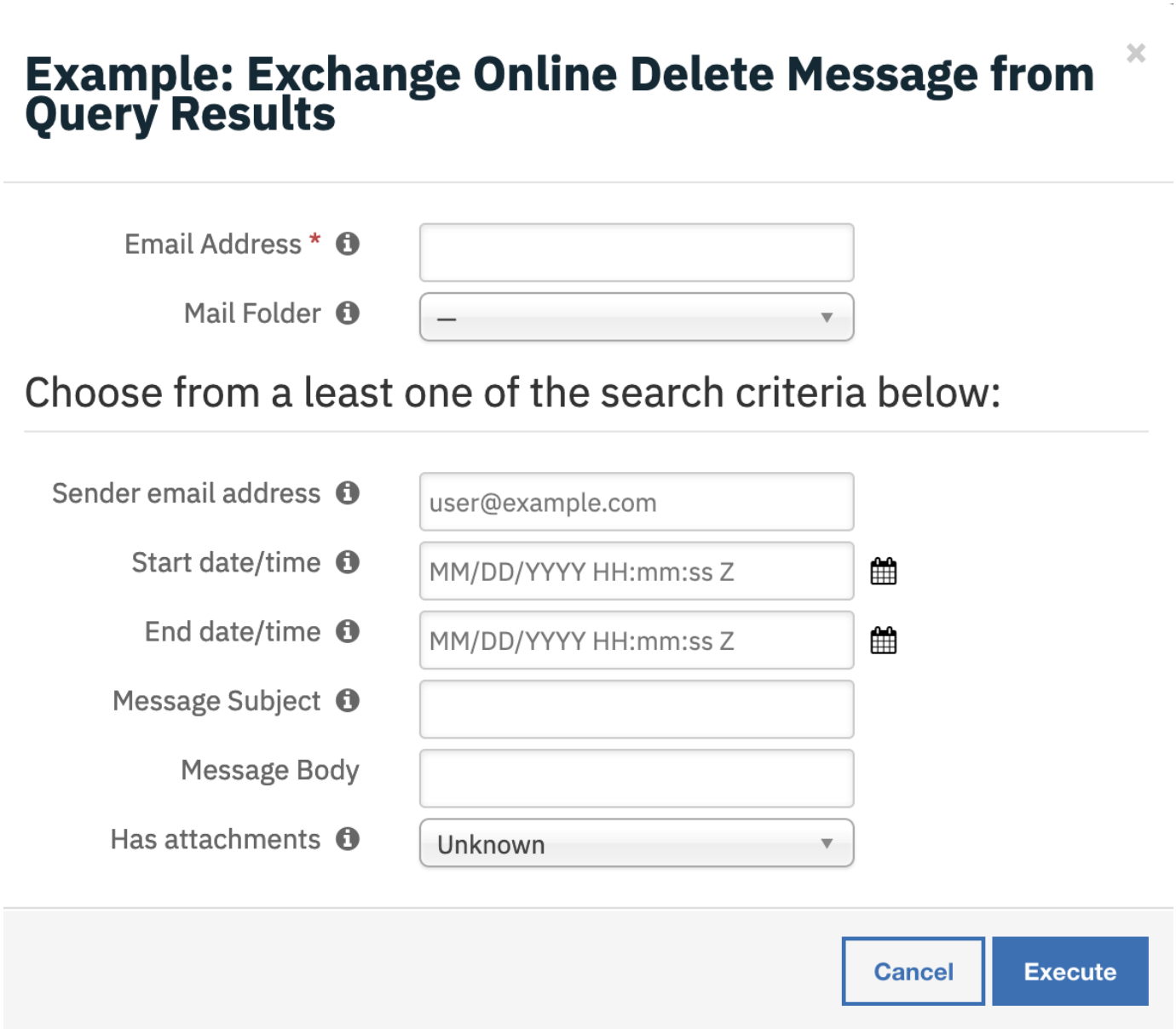
▶ Example Workflow Output:



▶ Example Rule:



The Example: Exchange Online Delete Message From Query Results incident menu item rule will bring up the rule activity popup dialog pictured below to prompt for input querying message to be deleted. See the Query function section for a description of querying.

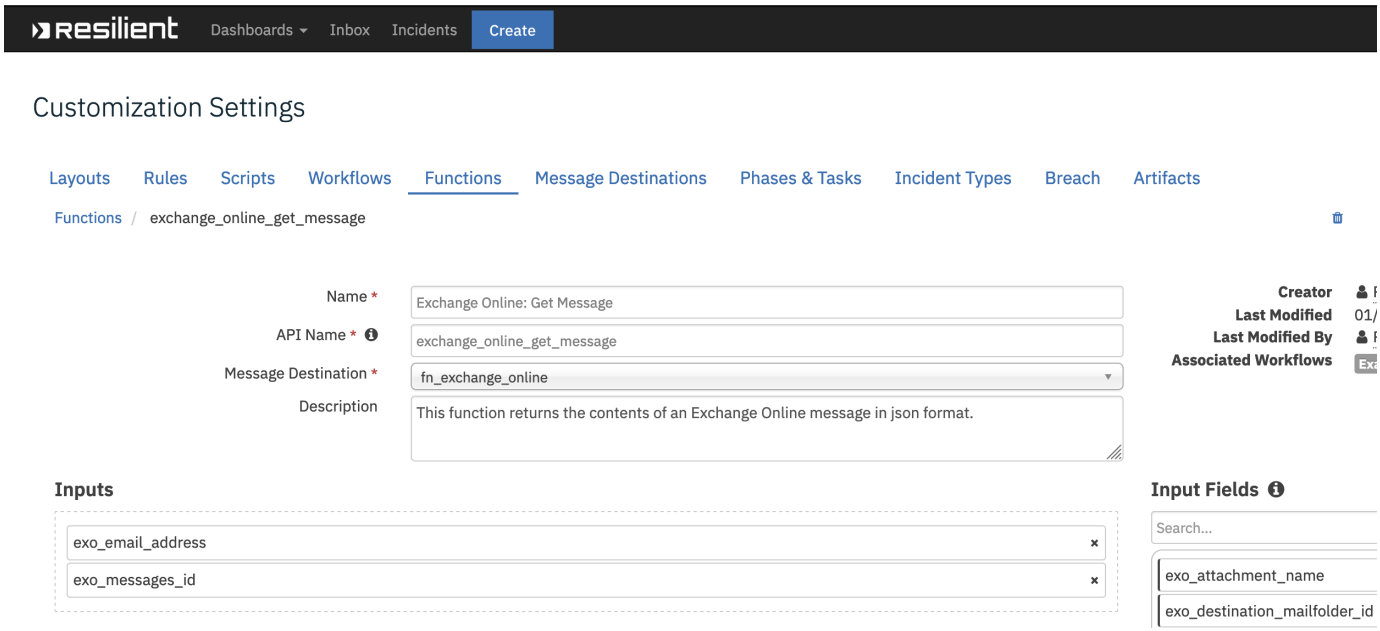


Function - Exchange Online: Get Message

Exchange Online: Get Message function requires the following Microsoft Graph API Application permission:

- **Mail.Read**

This function returns the contents of an Exchange Online message in JSON format.



▶ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_messages_id	text	Yes	-	The message ID of the message to get

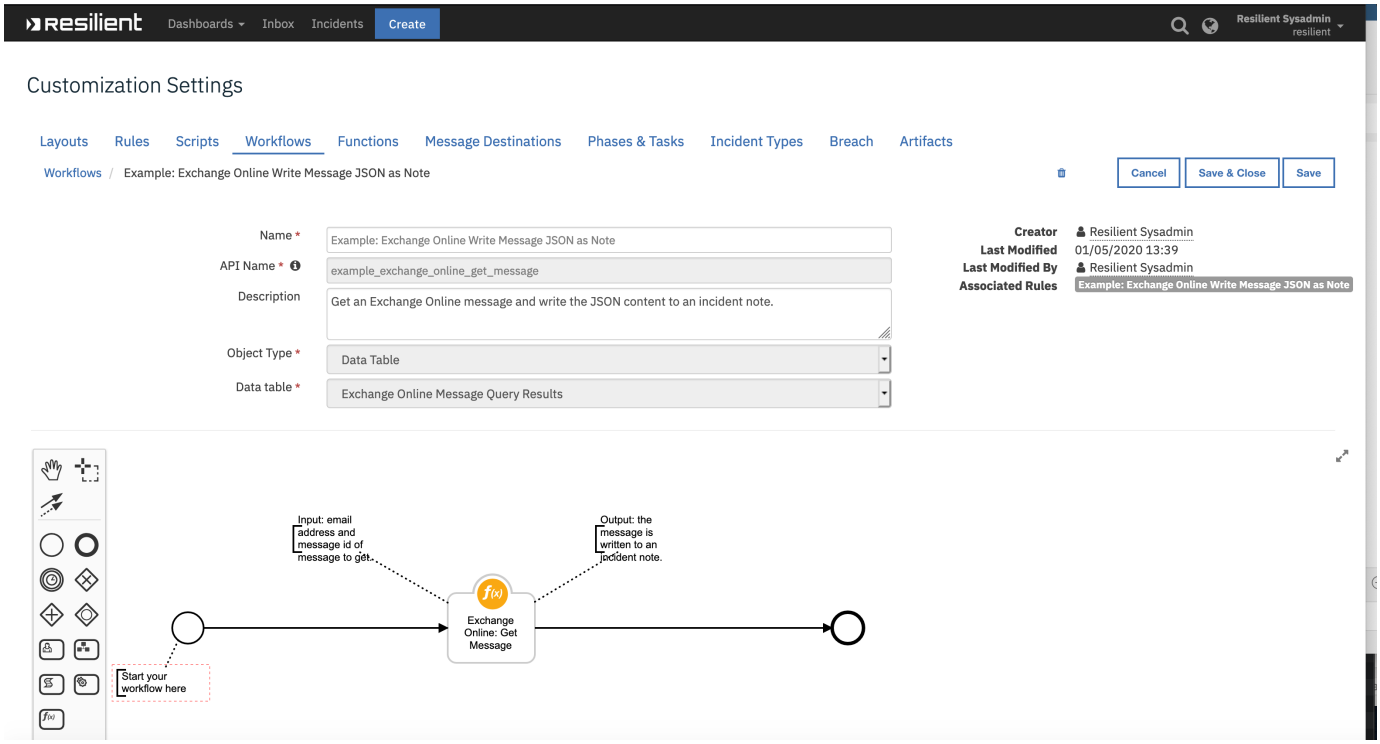
▶ Outputs:

```
results = {'inputs': {u'exo_messages_id':
u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
Jf-ANAwqcRJF4hFv_x44UAAAinByvAABJf-ANAwqcRJF4hFv_x44UAAAinIROAAA=',
                    u'exo_email_address':
u'resilient2@securitypocdemos.onmicrosoft.com'},
          'metrics': {'package': 'fn-exchange-online',
                      'timestamp': '2020-02-03 15:39:31',
                      'package_version': '1.0.0',
                      'host': 'cambridge.ibm.com',
                      'version': '1.0', '
                      'execution_time_ms': 654},
          'success': True,
          'pretty_string': u'{\n    "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#users(\'resilient2%40securitypocdemos.onmicro
soft.com\')/messages/$entity",\n    "@odata.etag":
"W/\\"CQAAABYAAABJf/ANAwqcRJF4hFv+x44UAAAil53/\\"",\n    "bccRecipients": [],\n
"body": {\n        "content": "<html>\\r\\n<head>\\r\\n<meta http-equiv=\\"Content-
Type\\" content=\\"text/html; charset=utf-8\\">\\r\\n<meta content=\\"text/html;
charset=us-ascii\\">\\r\\n</head>\\r\\n<body>\\r\\n<div class=\\"rte\\">\\r\\n<div>test
text area body</div>\\r\\n</div>\\r\\n</body>\\r\\n</html>\\r\\n",\n
"contentType": "html"\n    },\n    "bodyPreview": "test text area body",\n
"categories": [],\n    "ccRecipients": [],}.....',
          'content': {u'sentDateTime': u'2020-01-29T16:17:23Z',
```

```
u'conversationId':u'AAQkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAQAGjzfC_8ndROs
7O00o-q8ys=',
                    u'isDraft': False,
                    u'internetMessageId': u'<MMMMprod.outlook.com>',
                    u'id':
u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
Jf-ANAwqcRJF4hFv_x44UAAAinByvAABJf-ANAwqcRJF4hFv_x44UAAAinIROAAA=',
                    u'isReadReceiptRequested': False,
                    u'subject': u'test text area body',
                    u'lastModifiedDateTime': u'2020-02-03T16:46:41Z',
                    u'bodyPreview': u'test text area body',
                    u'from': {u'emailAddress': {u'name': u'Resilient User 3',
u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
                    u'flag': {u'flagStatus': u'notFlagged'},
                    .
                    .
                    .
                    },
            'reason': None,
            'version': '1.0'}
```

▶ Workflows:

Write Message as Attachment.



▶ Example Pre-Process Script:

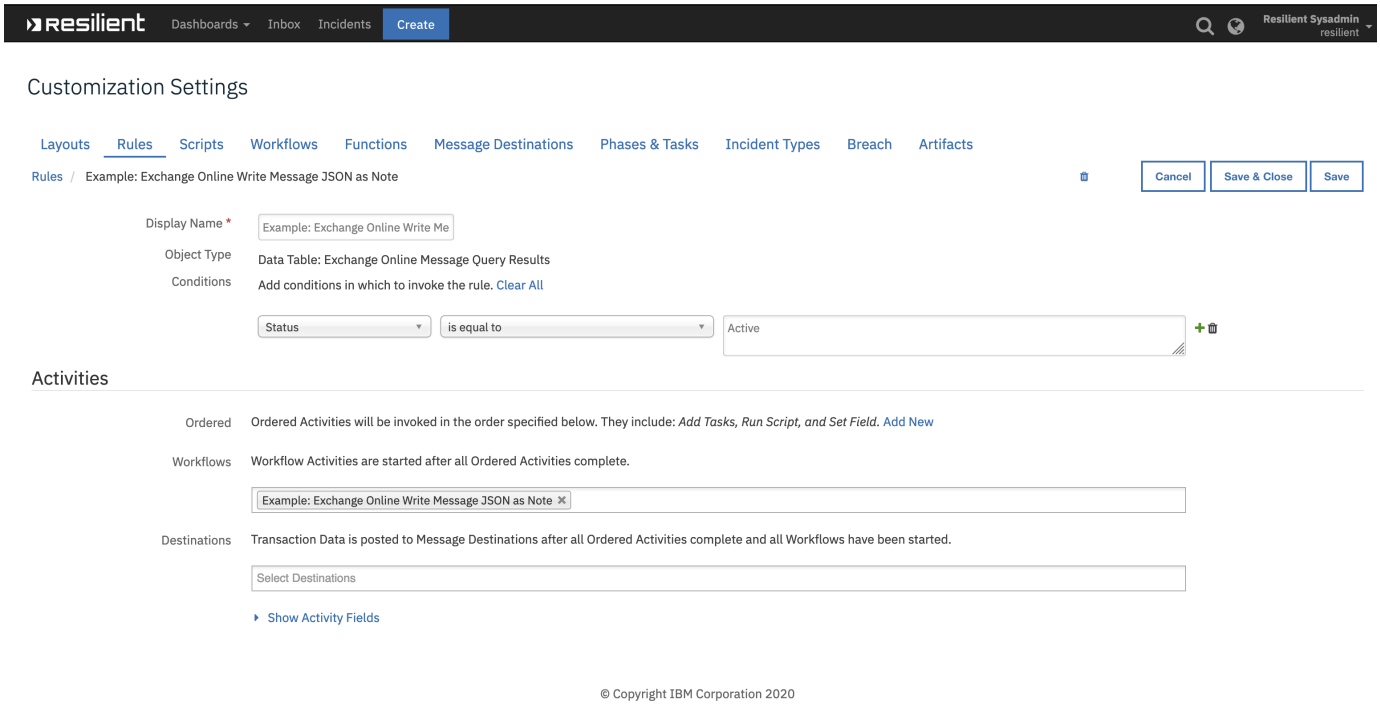
```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
```

▶ Example Post-Process Script:

```
# Print the message to an incident note if it is found, otherwise update the status as
Not Found in the datatable.
if results.content["error"] is not None:
```

```
  noteText = u"Exchange Online message NOT FOUND: \n email address: {0}\n message ID:
{1}\n{2}".format(results.inputs["exo_email_address"], results.inputs["exo_messages_id"],
results.pretty_string)
  row.exo_dt_status = "Not Found"
  row..exo_dt_web_link = ""
else:
  noteText = u"Exchange Online email address: {0}
message:\n{1}".format(results.inputs["exo_email_address"], results.pretty_string)

incident.addNote(noteText)
```

▶ Example Rule:

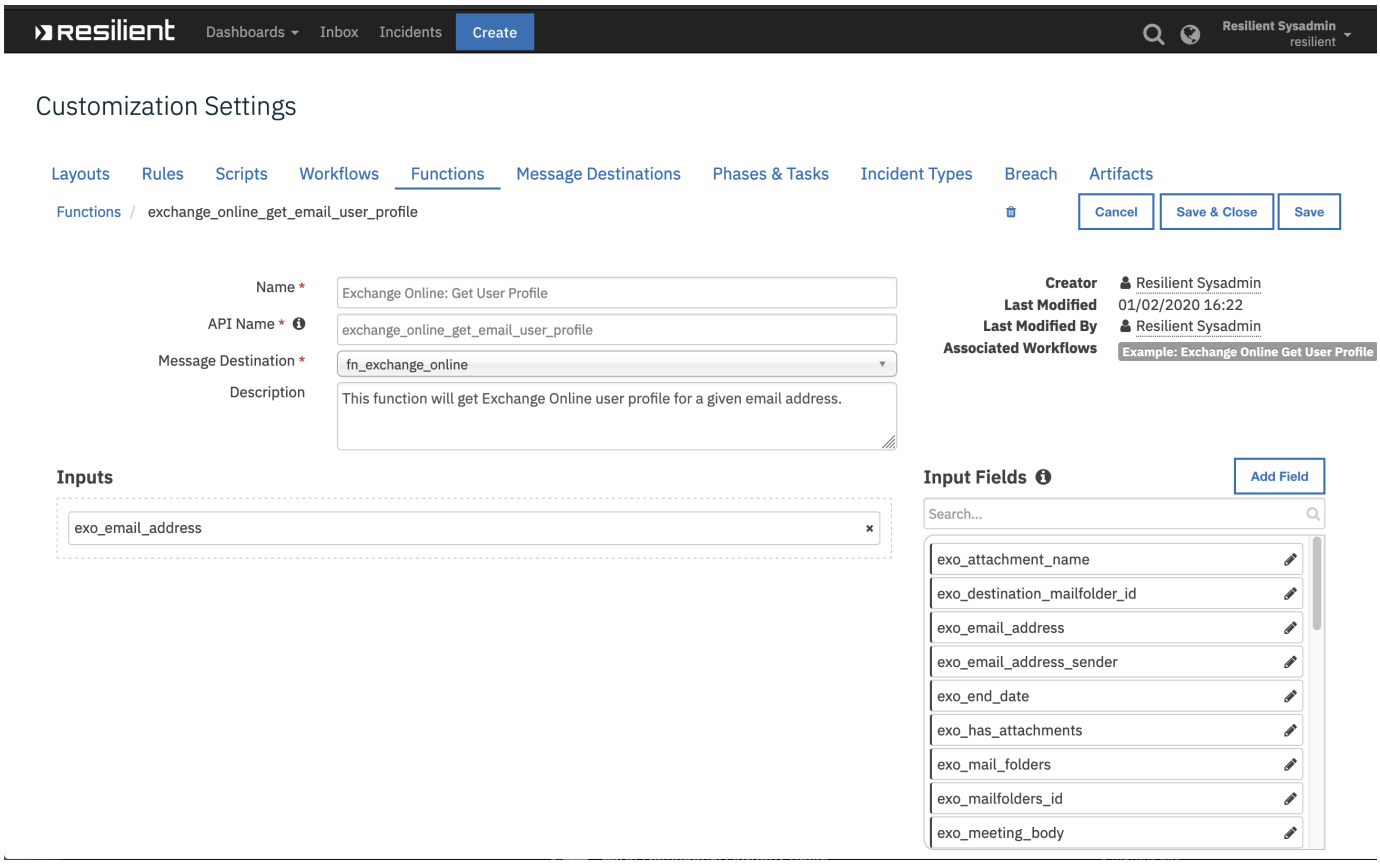


Function - Exchange Online: Get User Profile

Exchange Online: Get User Profile function requires the following Microsoft Graph API Application permission:

- **User.Read.All**

The Get User Profile function returns Exchange Online user profile for a given email address.



▶ Inputs:

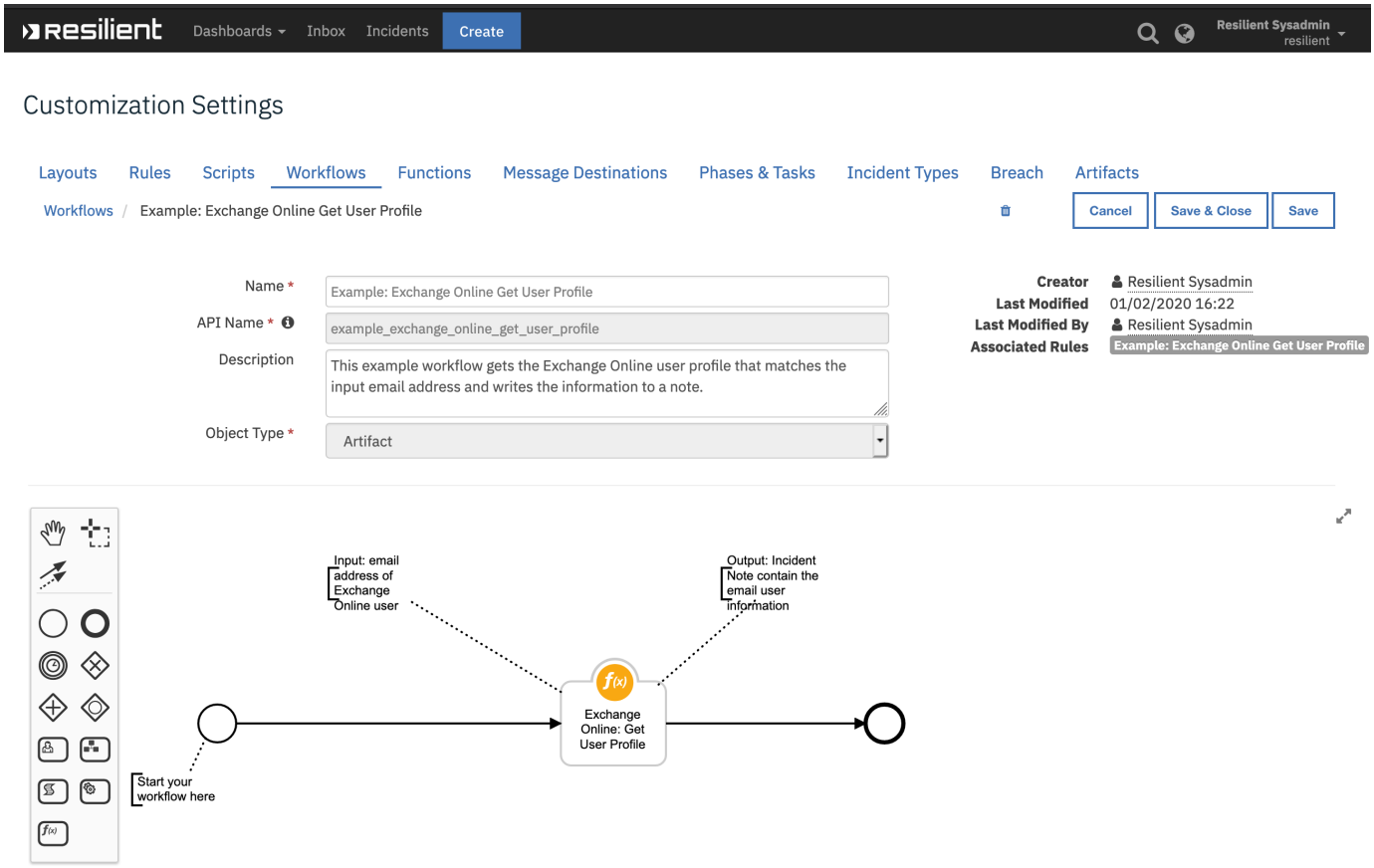
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account

▶ Outputs:

```
results = {
    'inputs': {u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},
    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-01-31 11:14:42',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 599},
    'success': True,
    'pretty_string': u'{\n    "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#users/$entity",\n    "businessPhones": [],\n
"displayName": "Resilient User 2",\n    "givenName": "Resilient User 2",\n    "id":
"393c1ebb-8222-4ba1-8665-f54eaf7f024f",\n    "jobTitle": null,\n    "mail":
"resilient2@securitypocdemos.onmicrosoft.com",\n    "mobilePhone": null,\n
"officeLocation": null,\n    "preferredLanguage": "en-US",\n    "surname": "Resilient
User 2",\n    "userPrincipalName": "resilient2@securitypocdemos.onmicrosoft.com"\n}',
    'content': {
        u'displayName': u'Resilient User 2',
        u'mobilePhone': None,
        u'preferredLanguage': u'en-US',
        u'jobTitle': None,
        u'userPrincipalName': u'resilient2@securitypocdemos.onmicrosoft.com',
        u'@odata.context': u'https://graph.microsoft.com/v1.0/$metadata#users/$entity',
        u'officeLocation': None,
        u'businessPhones': [],
        u'mail': u'resilient2@securitypocdemos.onmicrosoft.com',
        u'surname': u'Resilient User 2',
```

```
        u'givenName': u'Resilient User 2',
        u'id': u'393c1ebb-8222-4ba1-8665-f54eaf7f024f'},
    'raw': '{"displayName": "Resilient User 2", "mobilePhone": null, "preferredLanguage":
"en-US", "jobTitle": null, "userPrincipalName":
"resilient2@securitypocdemos.onmicrosoft.com", "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#users/$entity", "officeLocation": null,
"businessPhones": [], "mail": "resilient2@securitypocdemos.onmicrosoft.com", "surname":
"Resilient User 2", "givenName": "Resilient User 2", "id": "393c1ebb-8222-4ba1-8665-
f54eaf7f024f"}',
    'reason': None,
    'version': '1.0'
}
```


▶ Workflows:



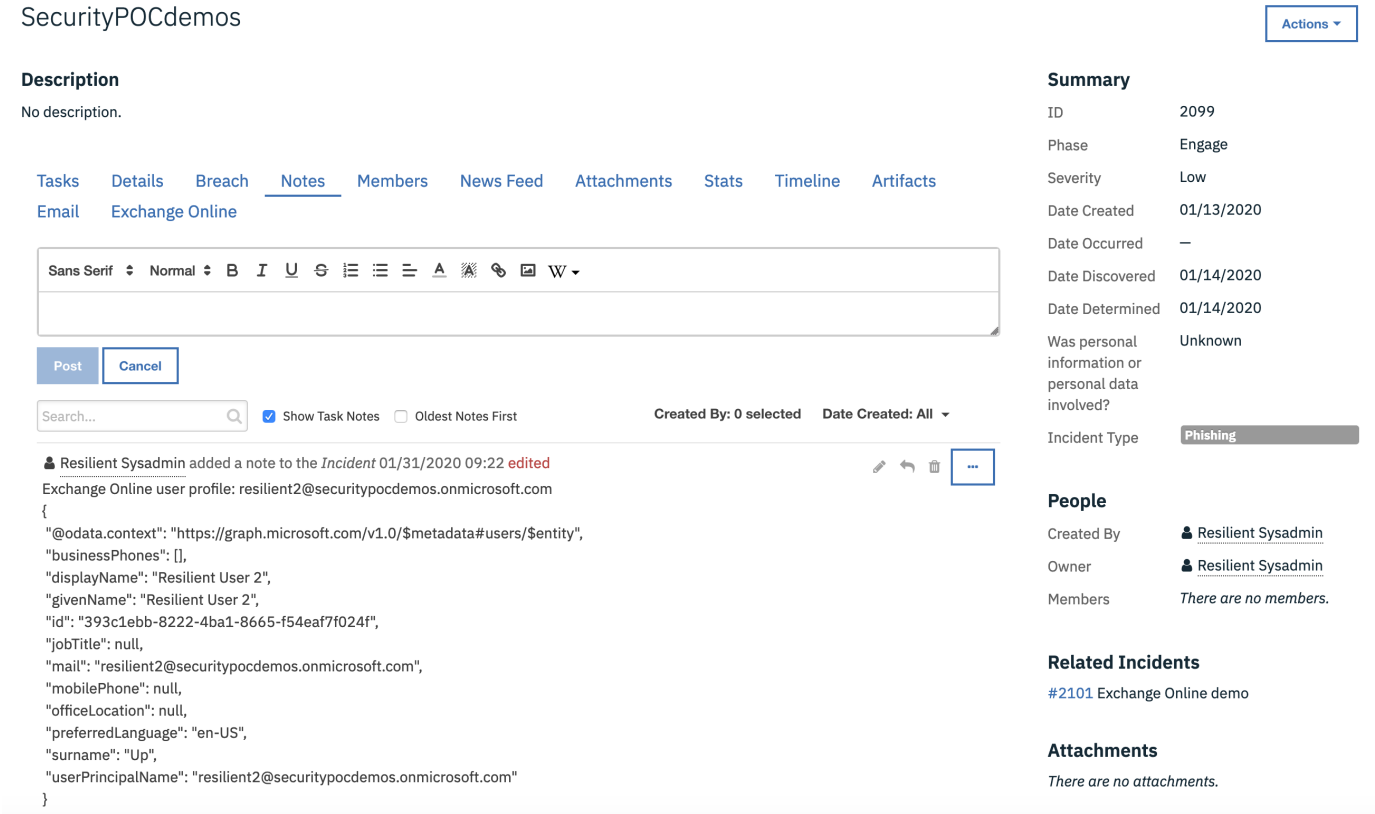
▶ Example Pre-Process Script:

```
inputs.exo_email_address = artifact.value
```

▶ Example Post-Process Script:

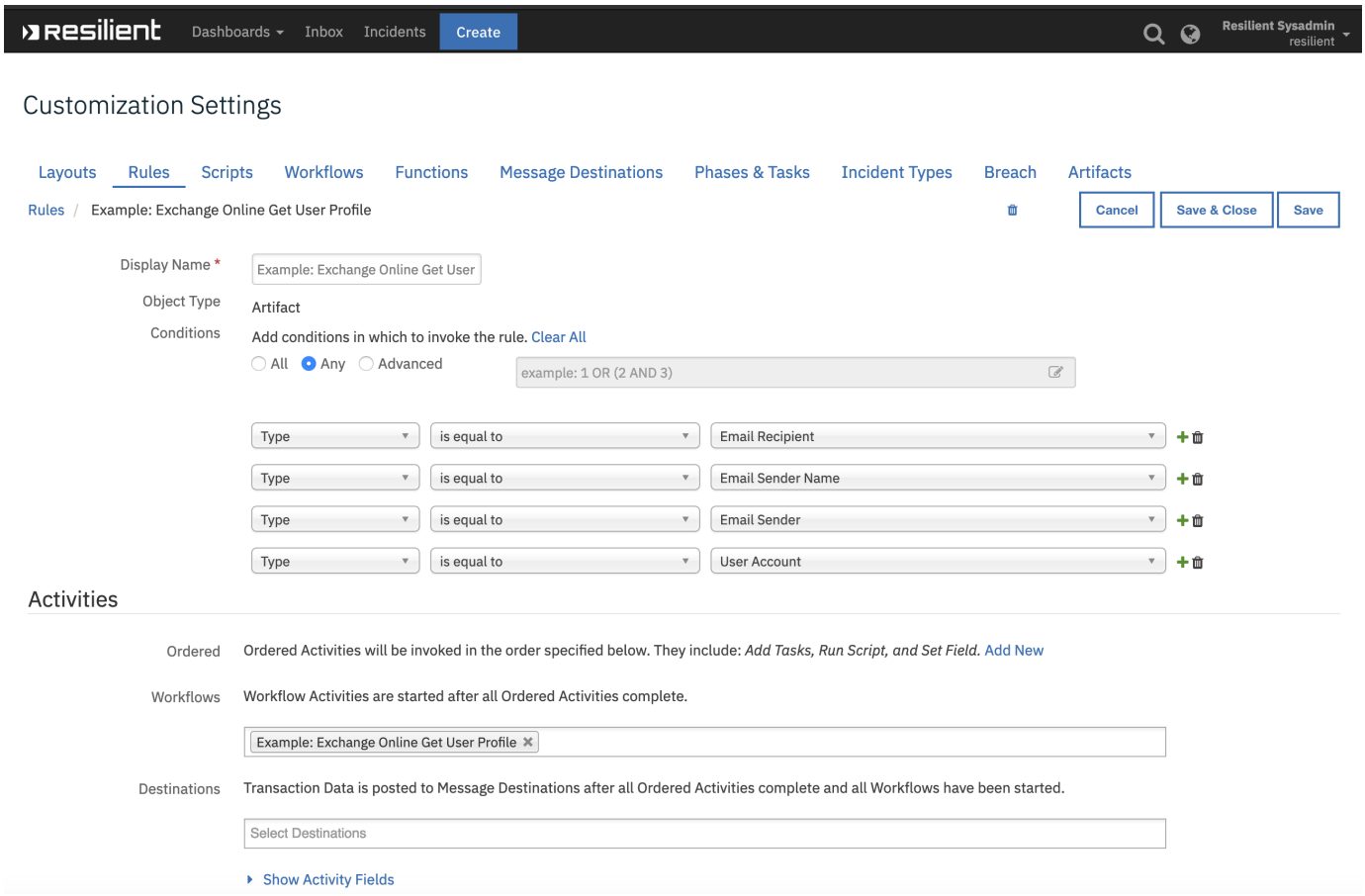
```
if results.content["error"] is not None:
  noteText = u"Exchange Online user profile NOT FOUND:
{0}\n{1}".format(results.inputs["exo_email_address"], results.pretty_string)
else:
  noteText = u"Exchange Online user profile:
{0}\n{1}".format(results.inputs["exo_email_address"], results.pretty_string)

incident.addNote(noteText)
```

▶ Example Workflow Output:



▶ Example Rule:

- Email Recipient
- Email Sender
- Email Sender Name
- User Account

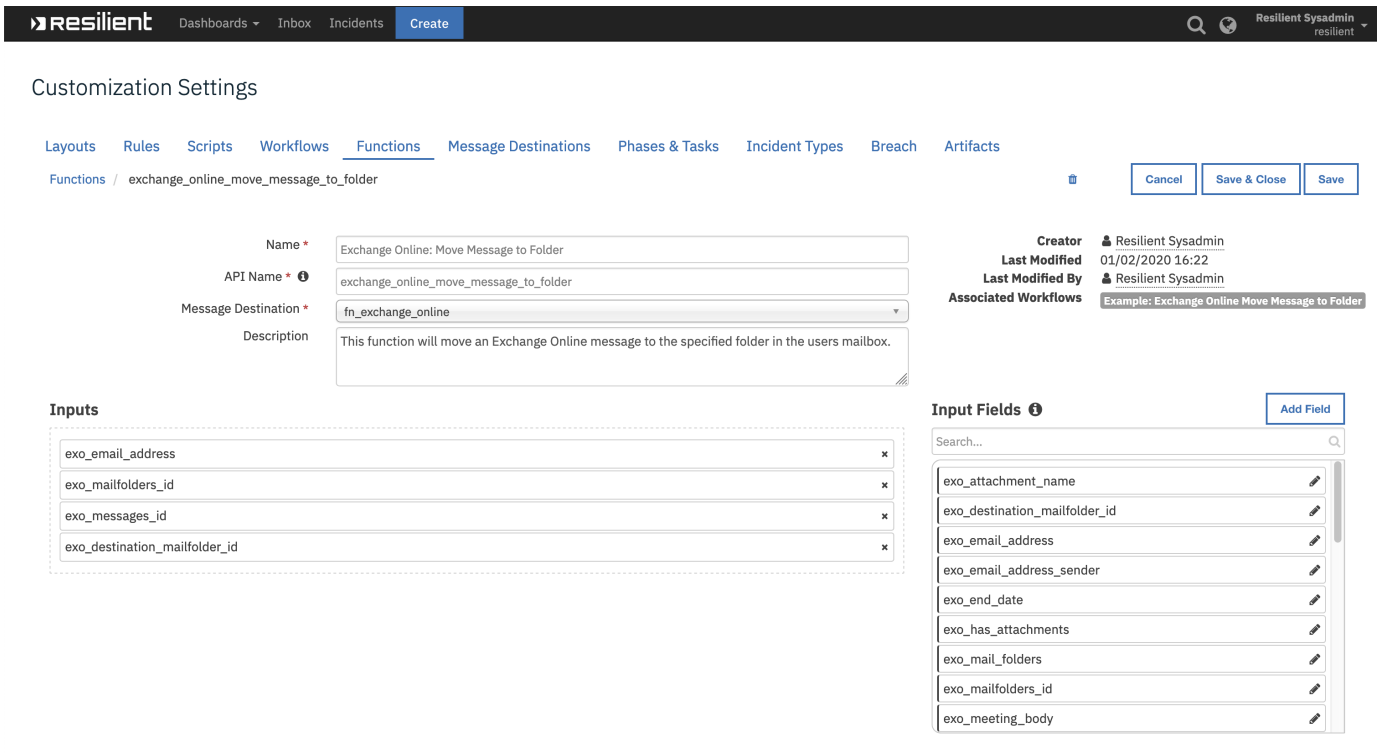


Function - Exchange Online: Move Message to Folder

Exchange Online: Move Message to Folder function requires the following Microsoft Graph API Application permission:

- **Mail.ReadWrite**

This function moves an Exchange Online message to the specified folder in the users mailbox.



▶ Inputs:

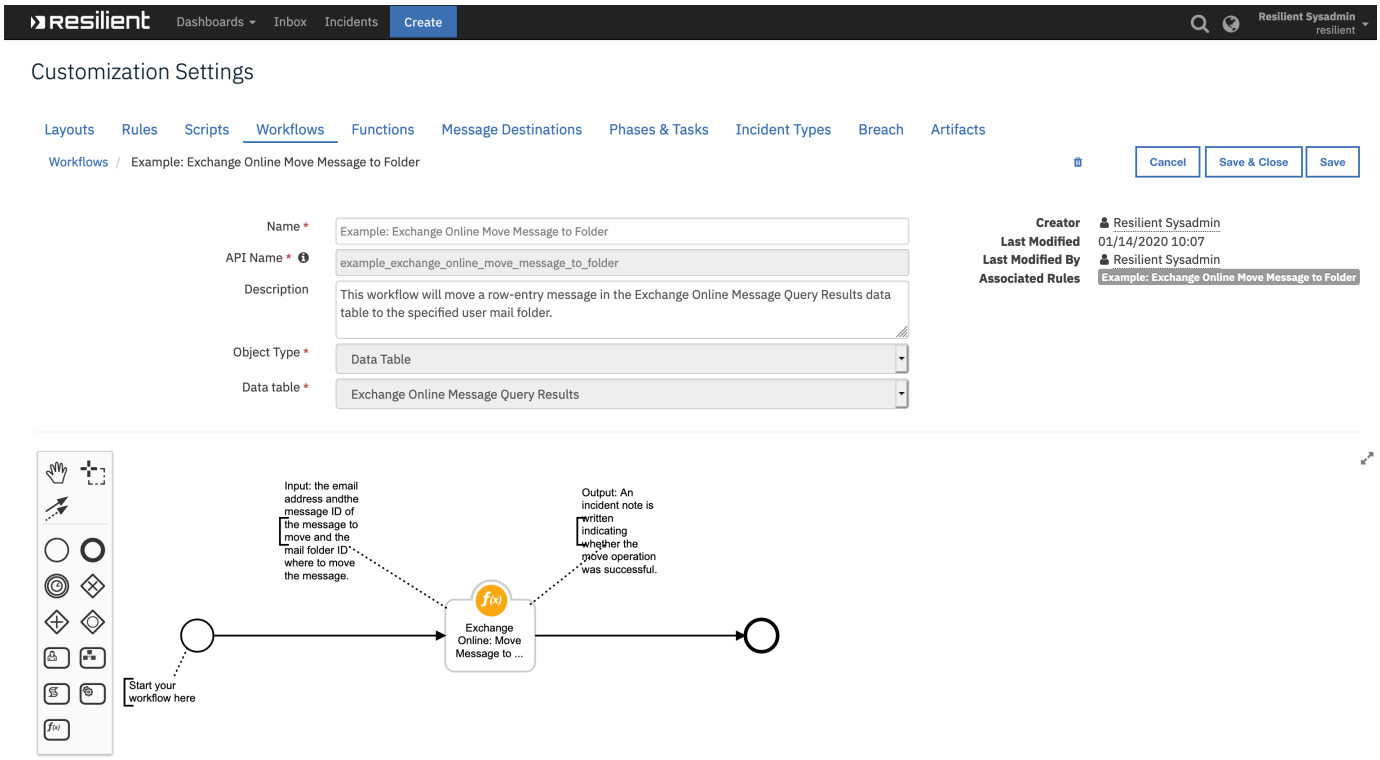
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_mailfolders_id	text	No	–	MailFolders ID
exo_messages_id	text	Yes	–	The message ID of the message to be deleted
exo_destination_mailfolder_id	select	Yes	recoverableitemsdeletions	Destination folder to which message is moved

▶ Outputs:


```
Result: {
  'inputs': {u'exo_destination_mailfolder_id': {u'id': 126,
                                                u'name': u'recoverableitemsdeletions'},
             u'exo_mailfolders_id': None,

u'exo_messages_id':u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEk
a4IVS4DBeEtMPuSEBwBJf-ANAwqcRJF4hFv_x44UAAAAAAEMAABJf-ANAwqcRJF4hFv_x44UAAAXaS2qAAA=',
u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},
  'metrics': {'package': 'fn-exchange-online',
              'timestamp': '2020-01-31 13:23:41',
              'package_version': '1.0.0',
              'host': 'MacBook-Pro.local',
              'version': '1.0',
              'execution_time_ms': 1706},
  'success': True,
  'content': {'value': True},
  'raw': '{"value": true}',
  'reason': None,
  'version': '1.0'}
```

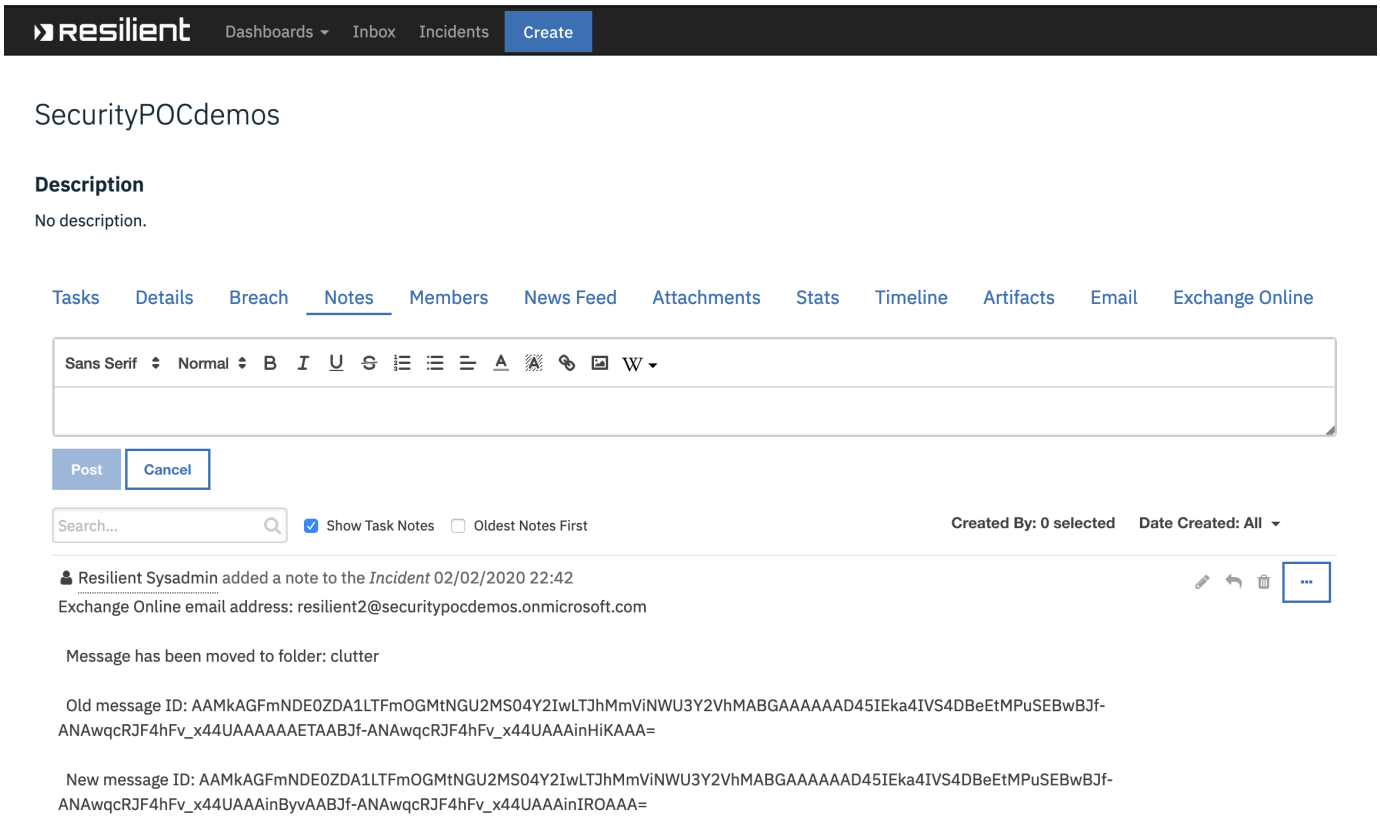

▶ Workflows:

- archive
- clutter
- conflicts
- conversationhistory
- deleteditems
- drafts
- inbox
- junkemail
- localfailures
- msgfolderroot
- outbox
- recoverableitemsdeletions
- scheduled
- searchfolders
- sentitems



▶ Example Workflow Output:

Below is a screen shot of an example Note after a message is moved to a folder:



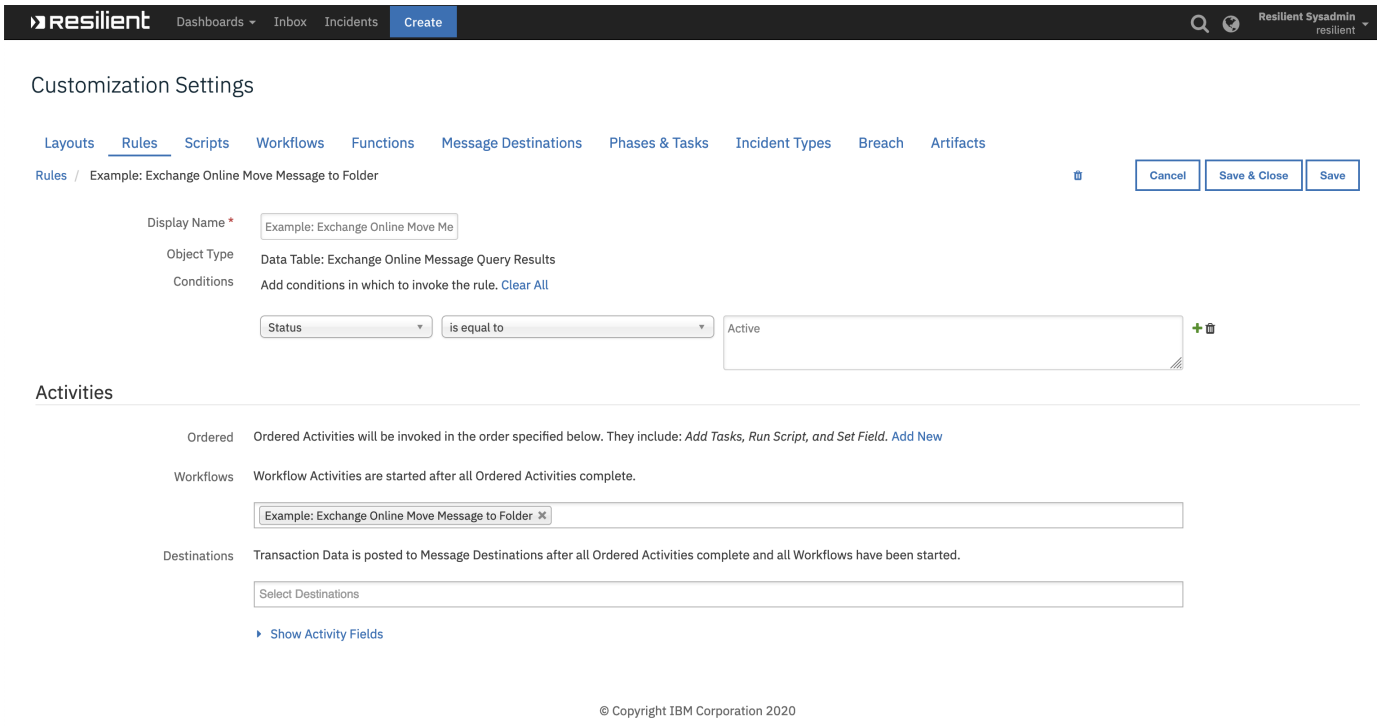
▶ Example Pre-Process Script:

```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_mailfolders_id = None
inputs.exo_messages_id = row.exo_dt_message_id
inputs.exo_destination_mailfolder_id = rule.properties.exo_destination_mailfolder_id
```

▶ Example Post-Process Script:

```
if results.content["error"] is not None:
  # Print the message to an incident note if it is found, otherwise update the status as
Not Found in the datatable.
  noteText = u"Exchange Online message NOT FOUND: \n email address: {0}\n message ID:
{1}".format(results.inputs["exo_email_address"], results.inputs["exo_messages_id"])
  status_text = u"""<p style= "color:{color}">{status} </p>""".format(color="red",
status="Not Found")
  row['exo_dt_status'] = helper.createRichText(status_text)
  row['exo_dt_web_link'] = ""
else:
  # When a message is moved it's ID changes, so update the new message ID into the data
table
  # The message status is still "Active" but the weblink is no longer valid, so make is
empty string.
  noteText = u"Exchange Online email address: {0}\n\n  Message has been moved to folder:
{1}\n\n  Old message ID: {2} \n\n  New message ID:
{3}".format(results.inputs["exo_email_address"],
results.inputs["exo_destination_mailfolder_id"]["name"],
results.inputs["exo_messages_id"], results.content["new_message_id"])
  row['exo_dt_message_id'] = results.content["new_message_id"]
  row['exo_dt_web_link'] = ref_html = u"""<a
href='{0}'>Link</a>""".format(results.content["new_web_link"])
incident.addNote(noteText)
```

▶ Example Rule:



Function - Exchange Online: Query Messages

Exchange Online: Query Messages function requires the following Microsoft Graph API Application permissions:

- **Mail.Read**
- **User.Read.All** (if querying the whole tenant)

The Exchange Online: Query Message function queries the Exchange Online to find messages matching the specified input parameters. The function returns a list of messages matching the search criteria.

The function will search over the following email accounts:

- all mailboxes of a tenant (specify "all", "ALL", "all users")
- single email address
- comma separated list of email addressed

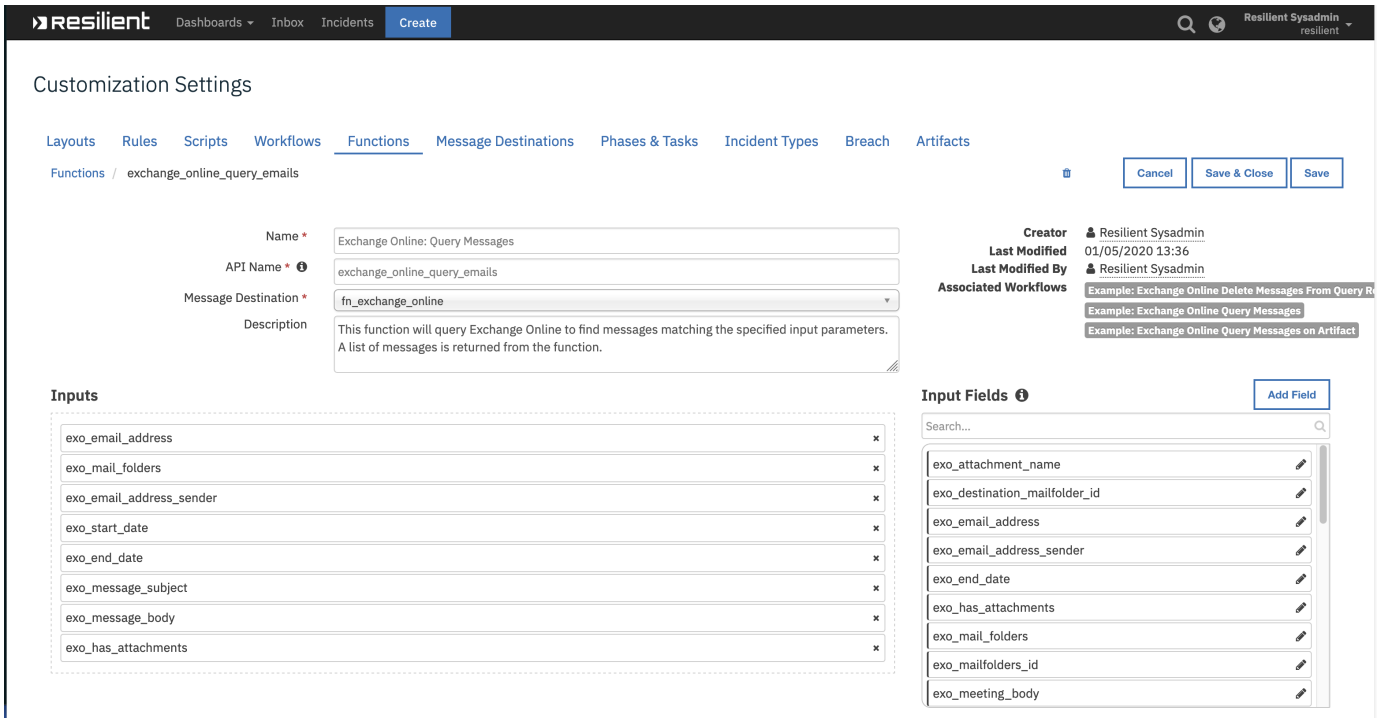
If no mail folder if specified, all folders and subdirectories are queried. The mail folder to be searched can be one of the list of Outlook "Well-known" folders:

- archive
- clutter
- conflicts
- conversationhistory
- deleteditems
- drafts
- inbox
- junkemail
- localfailures
- msgfolderroot
- outbox
- recoverableitemsdeletions
- scheduled
- searchfolders
- sentitems

At least one of the following search criteria must be passed to the query messages function:

- Sender email address
- Message received date/time start/end
- Message subject "contains" text
- Message body "contains" text
- Boolean flag indicating whether the message has an attachment

NOTE: There can be a large number of results of the Query Message function. If needed, use the max_user and max_messages parameters in the app.config file to limit the number of users searched and the number of messages returned from a query.



▶ Inputs:

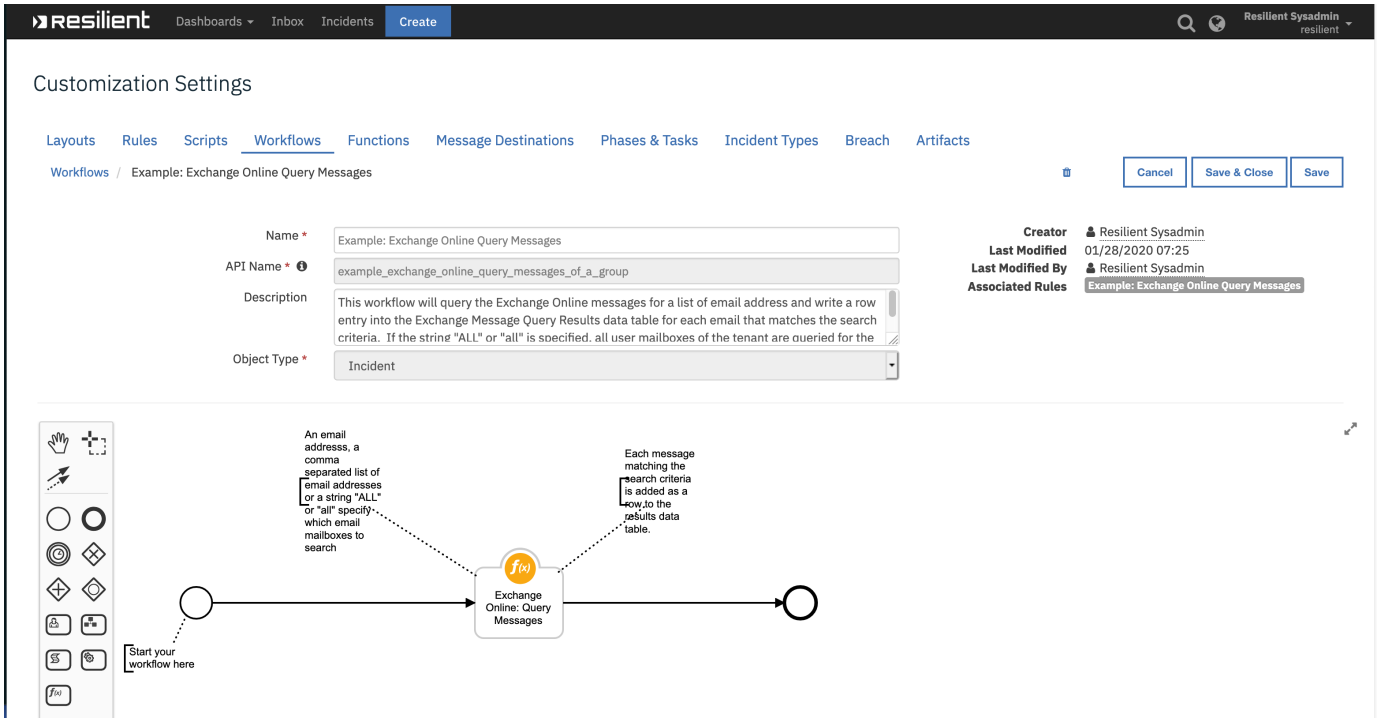
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_email_address_sender	text	No	user@example.com	Search messages sent from this email address; leave blank to ignore sender attribute
exo_end_date	datetimepicker	No	–	Query messages received ending at this date/time
exo_has_attachments	boolean	No	–	True to include attachments, False to exclude attachments, Unknown to get all
exo_mail_folders	text	No	Inbox	The folder to search in the users mailbox
exo_message_body	text	No	message body text	message body
exo_message_subject	text	No	message subject	message subject
exo_start_date	datetimepicker	No	–	Query messages received starting at this date/time

▶ Outputs:

```
results = {
   'inputs': {u'exo_start_date': None,
              u'exo_email_address_sender': None,
              u'exo_end_date': None,
              u'exo_message_subject': u'lunch',
              u'exo_has_attachments': None,
              u'exo_email_address': u'all',
```

```
                u'exo_mail_folders': None,
                u'exo_message_body': None},
     'metrics': {'package': 'fn-exchange-online',
                 'timestamp': '2020-02-04 15:36:12',
                 'package_version': '1.0.0',
                 'host': 'MacBook-Pro.local',
                 'version': '1.0',
                 'execution_time_ms': 9492},
     'success': True,
     'content': [{'status_code': 200,
                  'email_address': u'resilient2@securitypocdemos.onmicrosoft.com',
                  'email_list': [{u'sentDateTime': u'2020-02-04T20:44:02Z',
                  u'conversationId':
u'AAQkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAQANIxI-VmHPNIslAVFC4yWrI=',
                  u'internetMessageId':
u'<MWHPR2201MB11359DA0C07AF09AB319DD9FB1030@MWHPR2201MB1135.namprd22.prod.outlook.com>',
                  u'id':
u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
Jf-ANAwqcRJF4hFv_x44UAAAAAAEJAABJf-ANAwqcRJF4hFv_x44UAAAlbtF2AAA=',
u'isReadReceiptRequested': False,
                  u'subject': u'lunch',
                  u'lastModifiedDateTime': u'2020-02-04T20:44:04Z',
                  u'bodyPreview': u"Let's have lunch at 12!",
                  u'from': {u'emailAddress': {u'name': u'Resilient User 2',
                            u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
                  u'flag': {u'flagStatus': u'notFlagged'},
                  u'isDraft': False,
                  u'replyTo': [],
                  u'changeKey': u'CQAAABYAAABJf/ANAwqcRJF4hFv+x44UAAAlac/G',
u'receivedDateTime': u'2020-02-04T20:44:03Z',
                  u'parentFolderId':
u'AQMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAuAAAD_OSBJGuCFUuAwXhLTD7khAEASX-
wDQMKnESReIRb-seOFAAAAgEJAAAA',
                  u'body': {u'content': u'<html>\r\n<head>\r\n<meta http-equiv="Content-
Type" content="text/html; charset=utf-8">\r\n<meta content="text/html; charset=iso-8859-
1">\r\n<style type="text/css" style="display:none">\r\n<!--\r\np\r\n\t{margin-
top:0;\r\n\tmargin-bottom:0}\r\n-->\r\n</style>\r\n</head>\r\n<body dir="ltr">\r\n<div
style="font-family:Calibri,Arial,Helvetica,sans-serif; font-size:12pt;
color:rgb(0,0,0)">\r\nLet\'s have lunch at 12!<br>\r\n</div>\r\n</body>\r\n</html>\r\n',
u'contentType': u'html'},
                  u'isDeliveryReceiptRequested': False,
                  u'importance': u'normal',
                  u'toRecipients': [{u'emailAddress': {u'name': u'Resilient User 3',
                                     u'address':
u'resilient3@securitypocdemos.onmicrosoft.com'}}],
                  u'ccRecipients': [],
                  u'isRead': True,
                  u'categories': [],
                  u'sender': {u'emailAddress': {u'name': u'Resilient User 2',
                              u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
                  u'createdDateTime': u'2020-02-04T20:44:02Z',
                  u'webLink': u'https://outlook.office365.com/owa/?
ItemID=AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPu
SEBwBJf%2FANAwqcRJF4hFv%2Bx44UAAAAAAEJAABJf%2FANAwqcRJF4hFv%2Bx44UAAAlbtF2AAA%3D&exvsurl=
1&viewmodel=ReadMessageItem',
                  u'conversationIndex': u'AQHV25vV0jEj9WYc80iyUBUULjJasg==',
                  u'hasAttachments': False,
                  u'bccRecipients': [],
                  u'inferenceClassification': u'focused',
                  u'@odata.etag': u'W/"CQAAABYAAABJf/ANAwqcRJF4hFv+x44UAAAlac/G"'}]}],
       'reason': None,
       'version': '1.0'
}
```

▶ Workflows:



▶ Example Pre-Process Script:

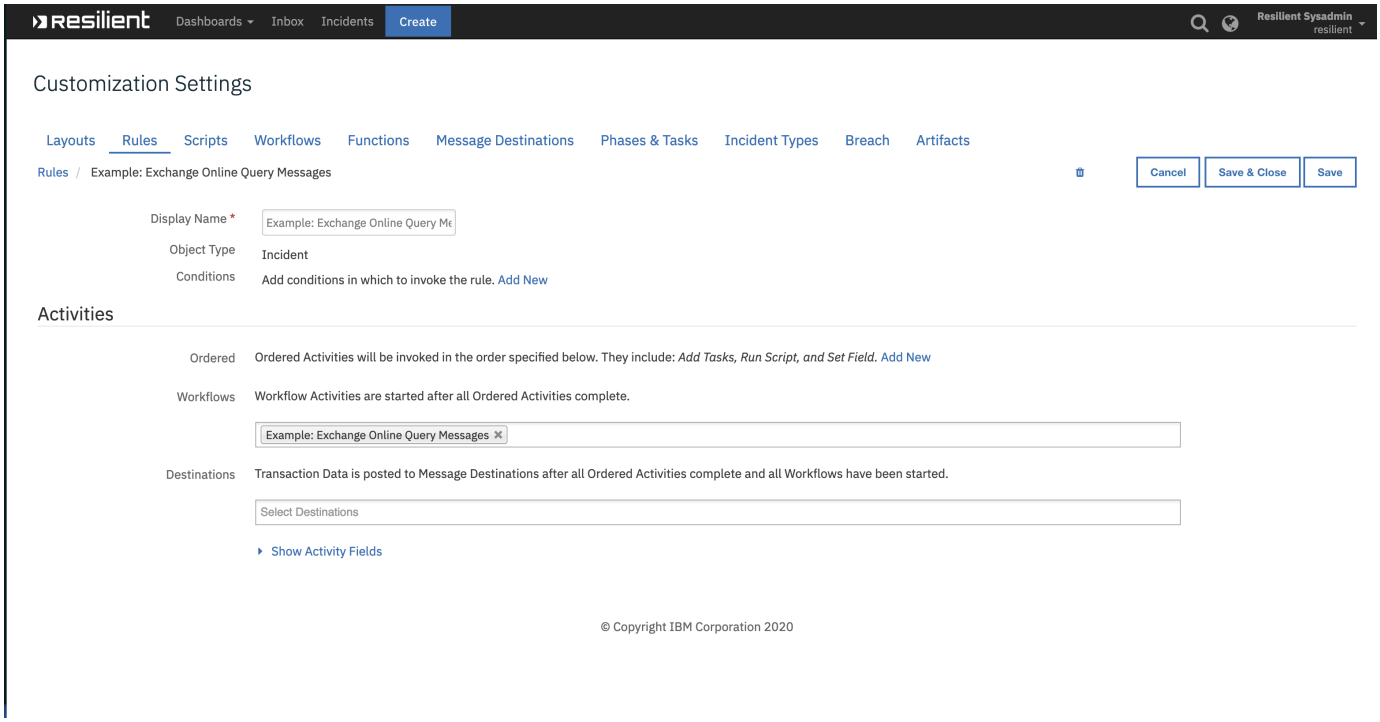
```
# Get the email address of the user whose mailbox will be queried.
inputs.exo_email_address = inputs.exo_email_address if
rule.properties.exo_email_address_list is None else
rule.properties.exo_email_address_list

# Get the search criteria from the activity rules if available.
inputs.exo_mail_folders         = inputs.exo_mail_folders         if
rule.properties.exo_mailfolder_id        is None else rule.properties.exo_mailfolder_id
inputs.exo_email_address_sender = inputs.exo_email_address_sender if
rule.properties.exo_email_address_sender is None else
rule.properties.exo_email_address_sender
inputs.exo_message_subject      = inputs.exo_message_subject      if
rule.properties.exo_message_subject      is None else rule.properties.exo_message_subject
inputs.exo_message_body         = inputs.exo_message_body         if
rule.properties.exo_message_body         is None else rule.properties.exo_message_body
inputs.exo_start_date           = inputs.exo_start_date           if
rule.properties.exo_start_date           is None else rule.properties.exo_start_date
inputs.exo_end_date             = inputs.exo_end_date             if
rule.properties.exo_end_date             is None else rule.properties.exo_end_date
inputs.exo_has_attachments      = inputs.exo_has_attachments      if
rule.properties.exo_has_attachments      is None else rule.properties.exo_has_attachments
```

▶ Example Post-Process Script:

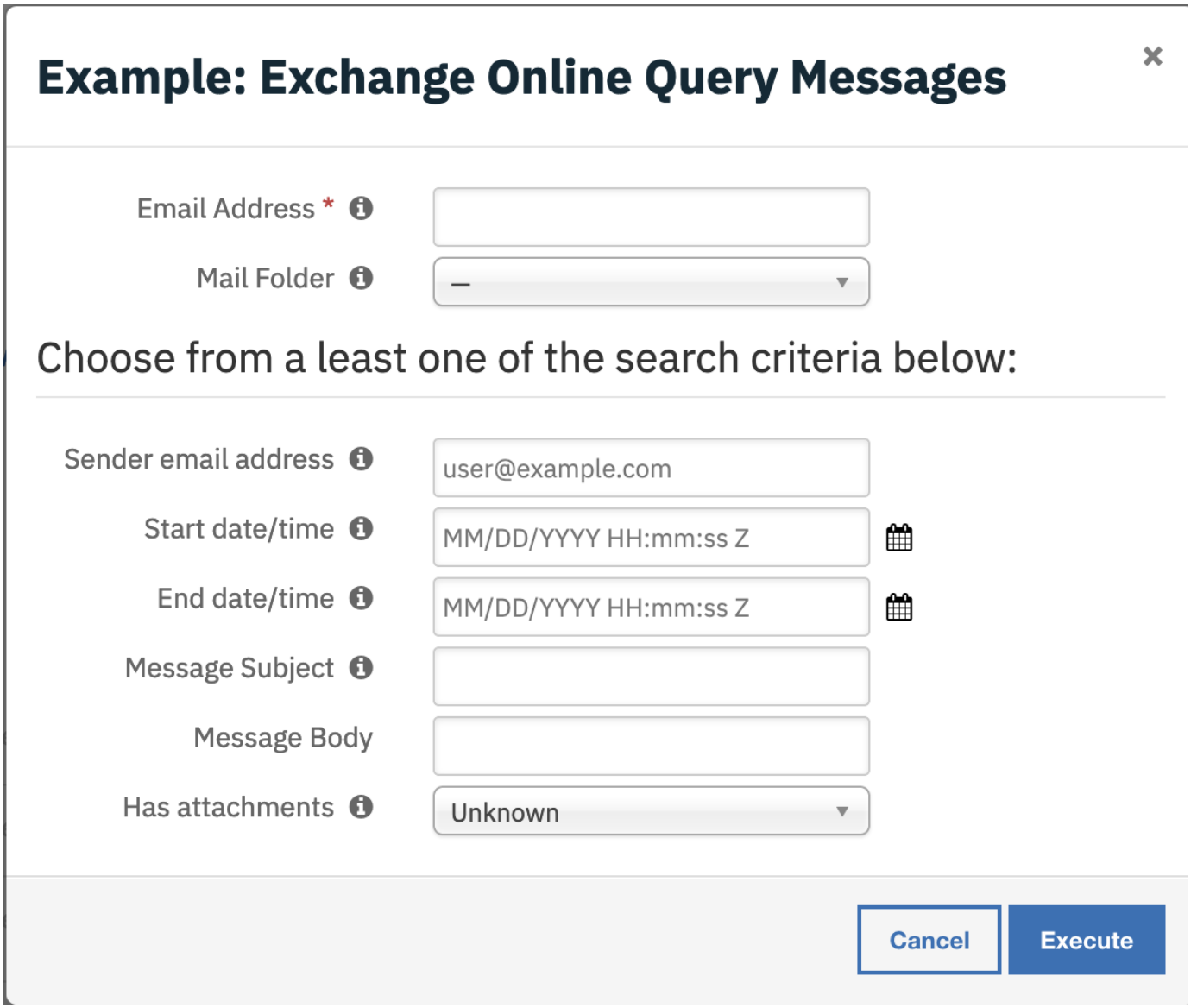
```
from java.util import Date

note = u"Exchange Online Query user:\n"
note_len = len(note)

# Add each email as a row in the query results data table
for user in results["content"]:
```

```
    # If an email address is not found post to a note.
    if user["status_code"] == 404:
      line = u"email address not found: {}\n".format(user["email_address"])
      note = note + line

    for email in user["email_list"]:
      message_row = incident.addRow("exo_message_query_results_dt")
      message_row.exo_dt_query_date = Date()
      message_row.exo_dt_message_id = email.id
      message_row.exo_dt_received_date   = email.receivedDateTime
      message_row.exo_dt_email_address = user["email_address"]
      if email.sender:
        message_row.exo_dt_sender_email = email.sender.emailAddress.address
      else:
        message_row.exo_dt_sender_email = ""
      message_row.exo_dt_message_subject = email.subject
      message_row.exo_dt_has_attachments = email.hasAttachments
      if email.webLink:
        ref_html = u"""<a href='{0}'>Link</a>""".format(email.webLink)
        message_row.exo_dt_web_link = helper.createRichText(ref_html)
      else:
        message_row.exo_dt_web_link = ""

      message_row.exo_dt_status = helper.createRichText("Active")

  # If any email addresses where not found post a note
  if len(note) > note_len:
    incident.addNote(note)
```

▶ Example Rule:



When the Example: Exchange Online Delete Messages from Query Results rule is activated, the following rule activity popup dialog prompts for input to create the meeting and send a message to the invitees:

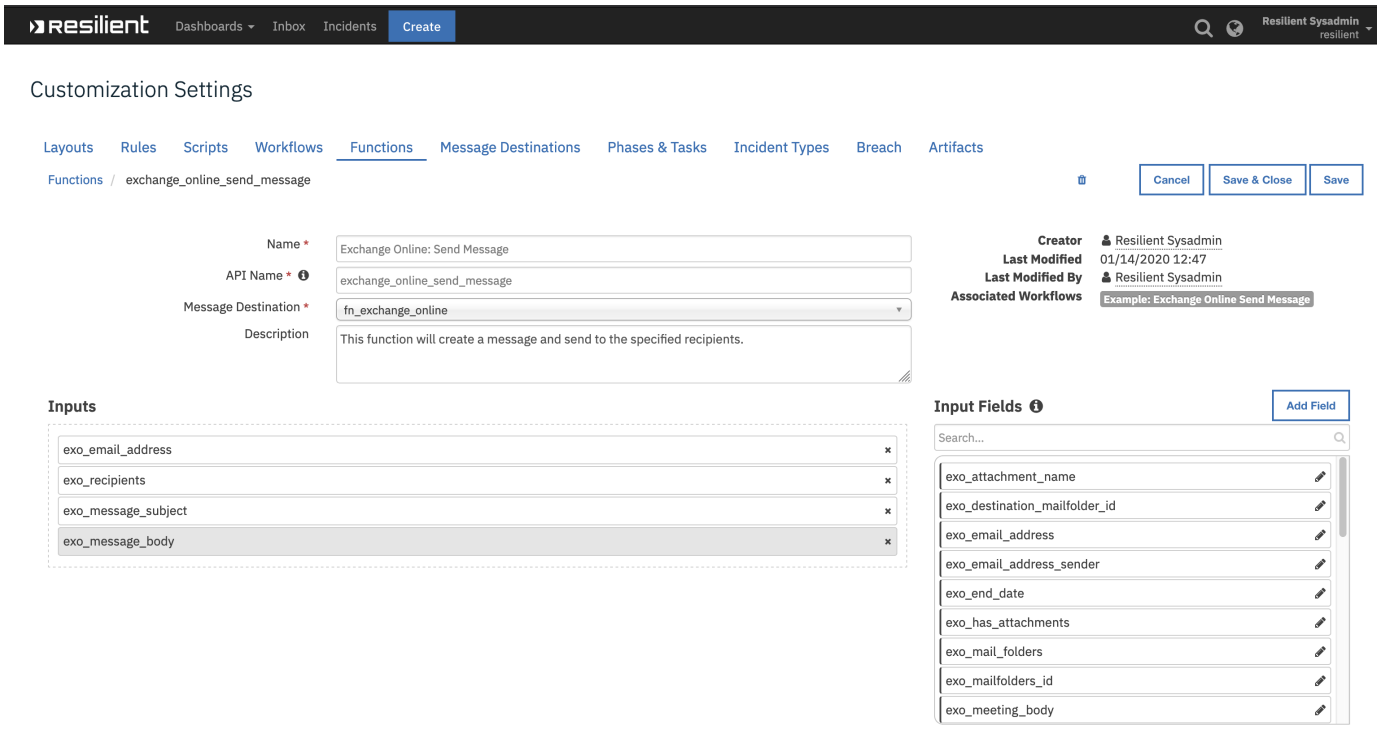


Function - Exchange Online: Send Message

Exchange Online: Send Message function requires the following Microsoft Graph API Application permission:

- **Mail.Send**

This function creates a message and sends it to the specified recipients.



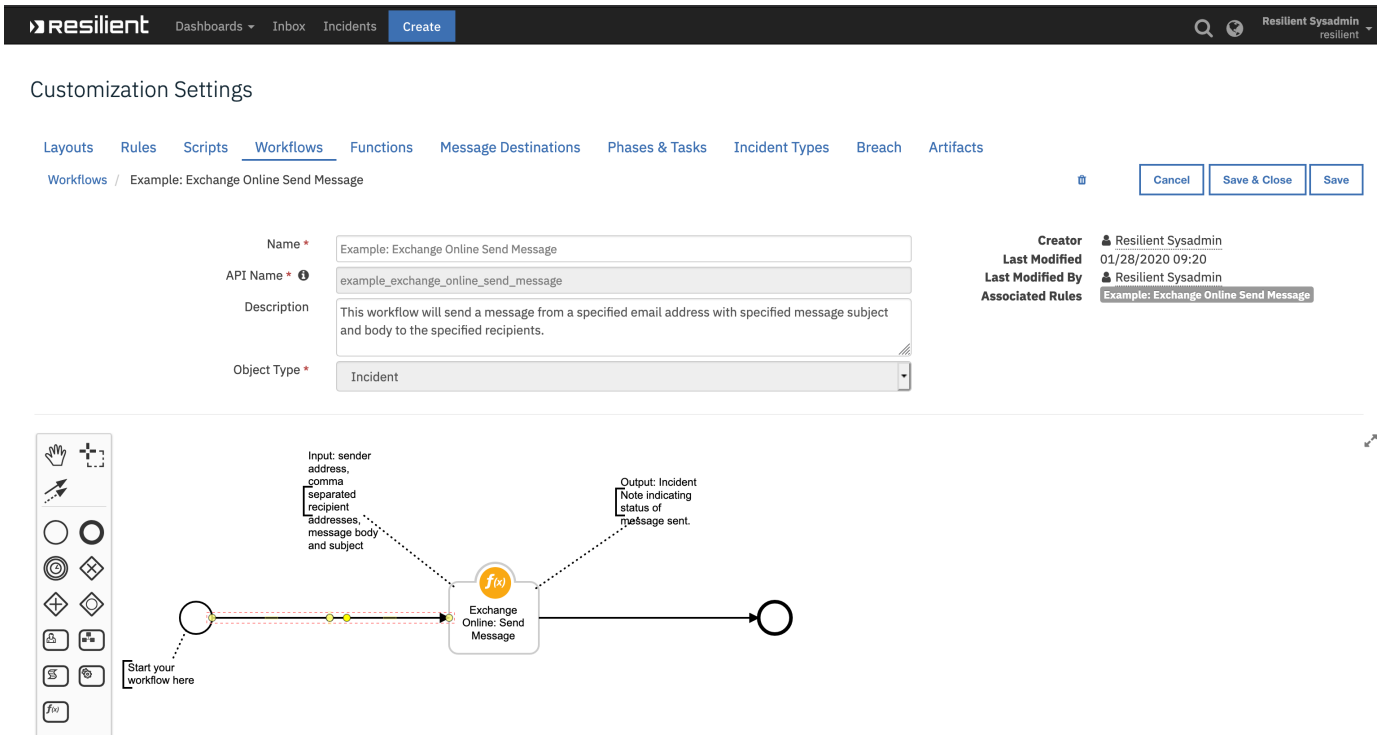
▶ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_message_body	text	No	message body text	Message body
exo_message_subject	text	No	message subject	Message subject
exo_recipients	text	Yes	–	Comma separated list of message recipients

▶ Outputs:

```
results = {
    'inputs': {u'exo_recipients': u'resilient2@securitypocdemos.onmicrosoft.com,
resilient3@securitypocdemos.onmicrosoft.com',
               u'exo_message_subject': u'Please investigate',
               u'exo_message_body': u'<div class="rte"><div>Can you look into this?</div>
<div><br /></div><div>Thanks!</div></div>',
               u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},

    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-02-04 11:18:16',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 796},
    'success': True,
    'content': {'value': True},
    'raw': '{"value": true}',
    'reason': None,
    'version': '1.0'}
```

▶ Workflows:



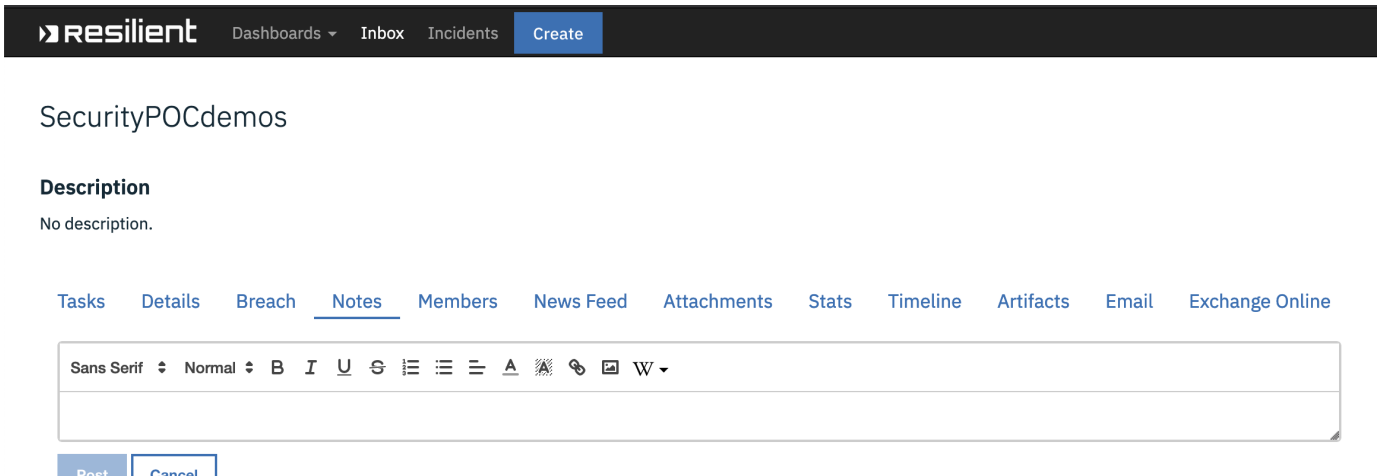
▶ Example Pre-Process Script:

```
inputs.exo_email_address   = inputs.exo_email_address   if
rule.properties.exo_message_sender_address is None else
rule.properties.exo_message_sender_address
inputs.exo_recipients      = inputs.exo_recipients      if
rule.properties.exo_message_recipients     is None else
rule.properties.exo_message_recipients
inputs.exo_message_subject = inputs.exo_message_subject if
rule.properties.exo_message_subject is None else rule.properties.exo_message_subject
inputs.exo_message_body    = inputs.exo_message_body    if
rule.properties.exo_message_send_body.content is None else
rule.properties.exo_message_send_body.content
```

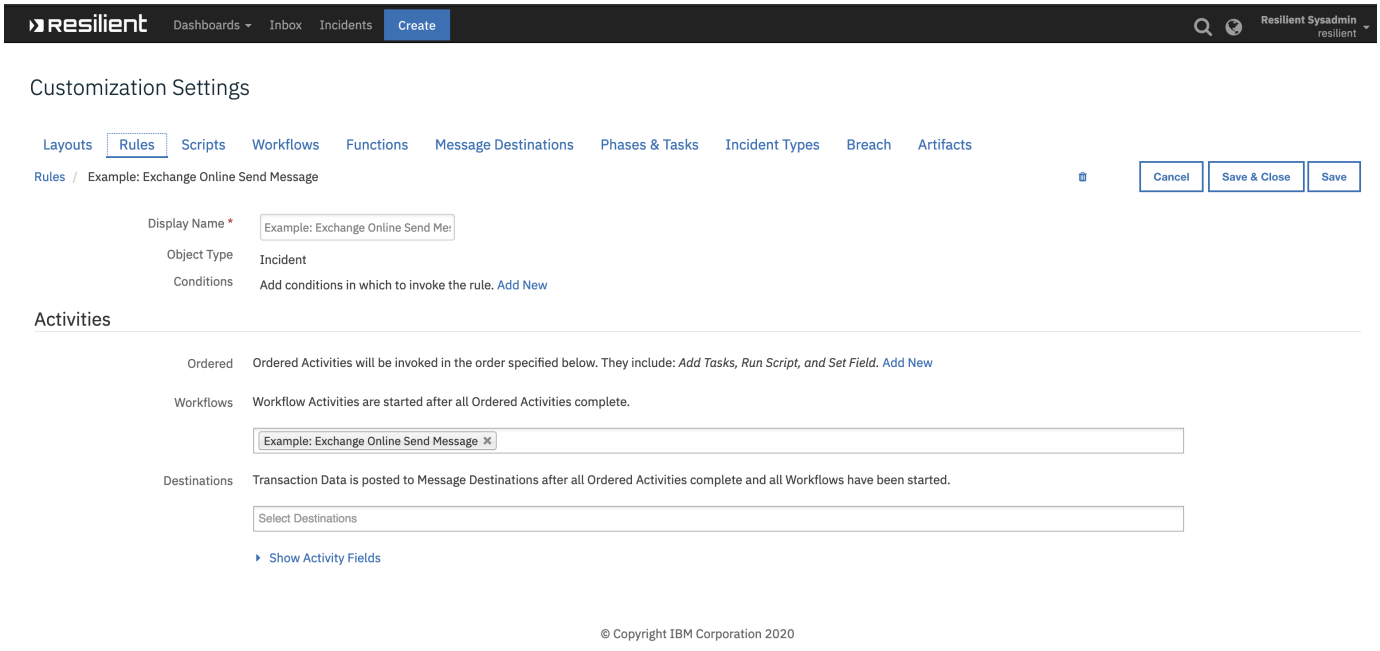
▶ Example Post-Process Script:

```
if results.success:
  noteText = u"Exchange Online message sent\n   From: {0}\n   To: {1}\n   Subject: {2}\n
Body: {3}".format(results.inputs["exo_email_address"], results.inputs["exo_recipients"],
results.inputs["exo_message_subject"], results.inputs["exo_message_body"])
else:
  noteText = u"Exchange Online message NOT sent\n   From: {0}\n  To:
{1}".format(results.inputs["exo_email_address"], results.inputs["exo_recipients"])

incident.addNote(noteText)
```

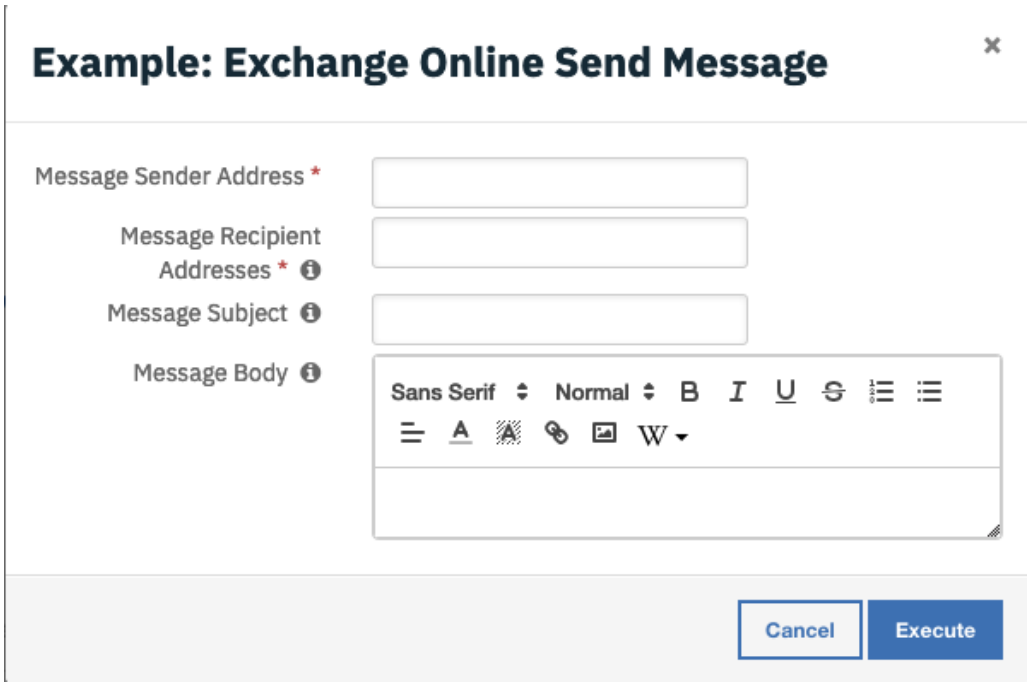
▶ Example Workflow Output:



▶ Example Rule:



When the Example Send Message rule is initiated the following rule activity popup dialog will appear prompting for input on the message to send:

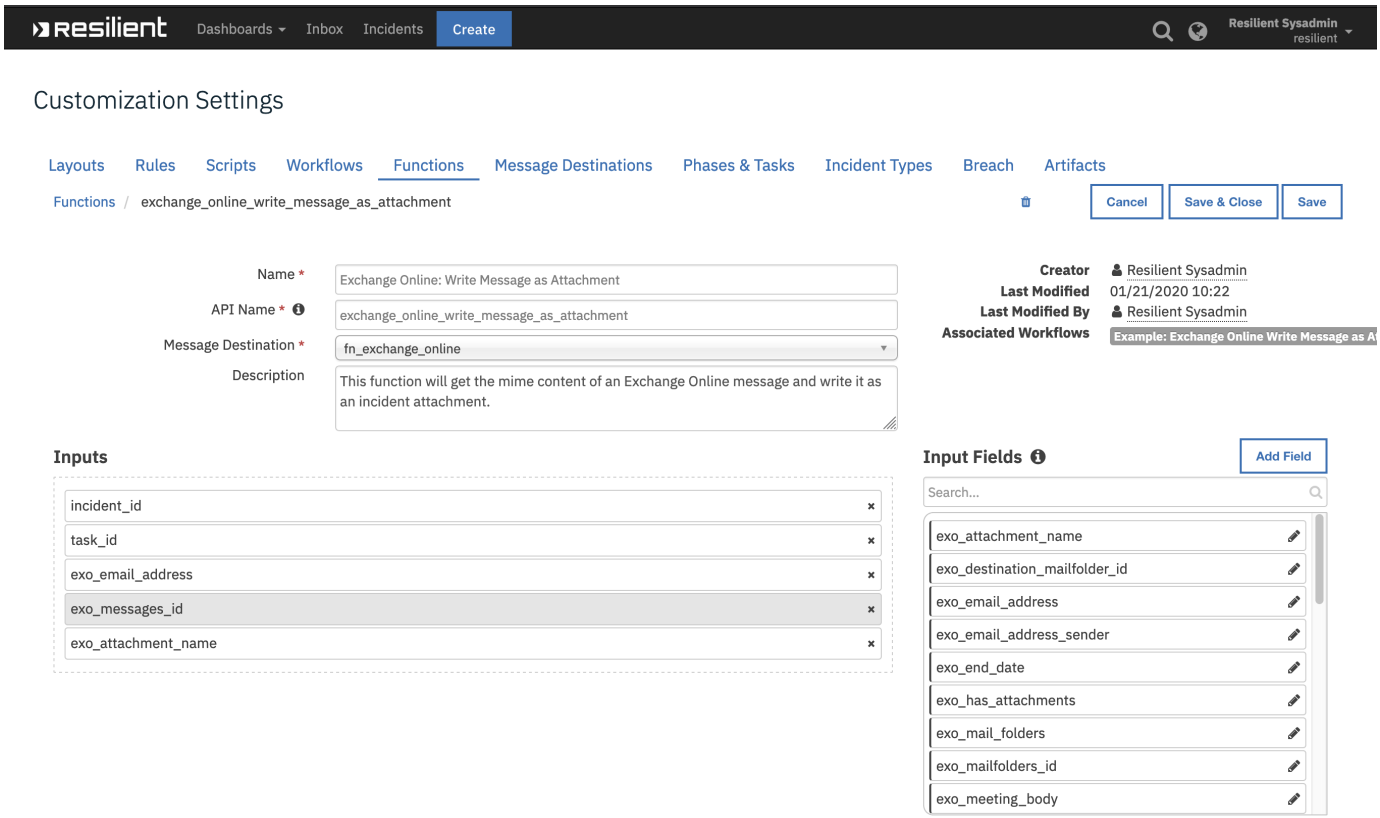


Function - Exchange Online: Write Message as Attachment

Exchange Online: Write Message as Attachment function requires the following Microsoft Graph API Application permission:

- **Mail.Read**

This function gets the mime content of an Exchange Online message and writes it as an incident attachment. The attachment file name is an optional parameter. The function uses a default message-{email-address}-{message-ID}.eml filename if none is specified.



▶ Inputs:

Name	Type	Required	Example	Tooltip

Name	Type	Required	Example	Tooltip
incident_id	number	Yes	-	-
task_id	number	No	-	-
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_messages_id	text	Yes	-	The message ID of the message to be deleted
exo_attachment_name	text	No	my-message.eml	The attachment file to which message is written

▶ Outputs:

```
results = {'inputs': {u'incident_id': 2099,
                      u'exo_attachment_name': u'my-message.eml',
                      u'exo_messages_id':
u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
Jf-ANAwqcRJF4hFv_x44UAAAinByvAABJf-ANAwqcRJF4hFv_x44UAAAinIROAAA=',
                       u'exo_email_address':
u'resilient2@securitypocdemos.onmicrosoft.com'},
           'metrics': {'package': 'fn-exchange-online',
                       'timestamp': '2020-02-03 14:54:13',
                       'package_version': '1.0.0',
                       'host': 'annmarie-mbp.cambridge.ibm.com',
                       'version': '1.0', 'execution_time_ms': 5929},
            'success': True, 'content': {'attachment_name': u'my-message.eml'},
            'raw': '{"attachment_name": "my-message.eml"}',
            'reason': None,
            'version': '1.0'}
}
```

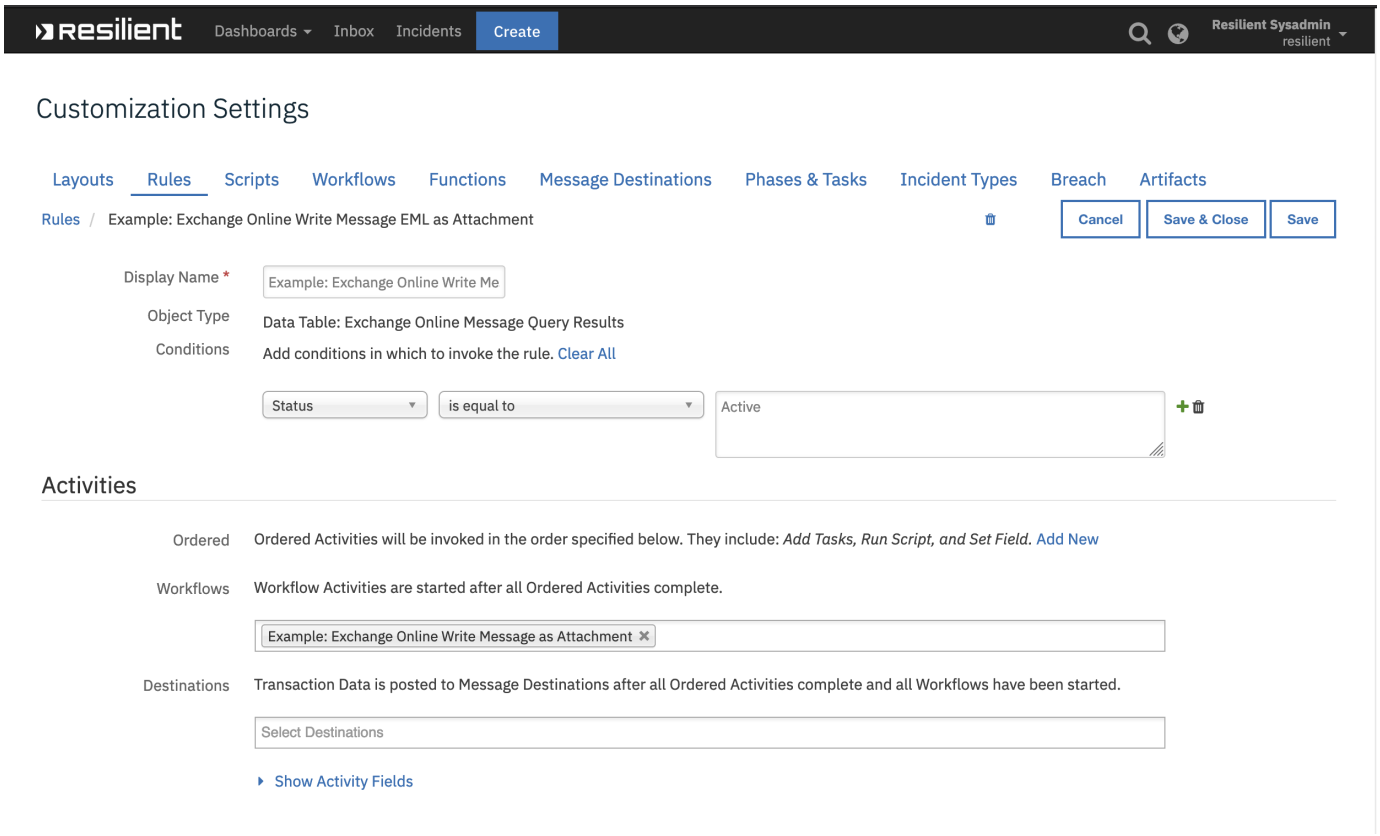
▶ Workflows:

```
inputs.incident_id = incident.id
#inputs.task_id = task.id
inputs.exo_attachment_name = rule.properties.exo_attachment_name
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
```

▶ Example Post-Process Script:

```
None
```

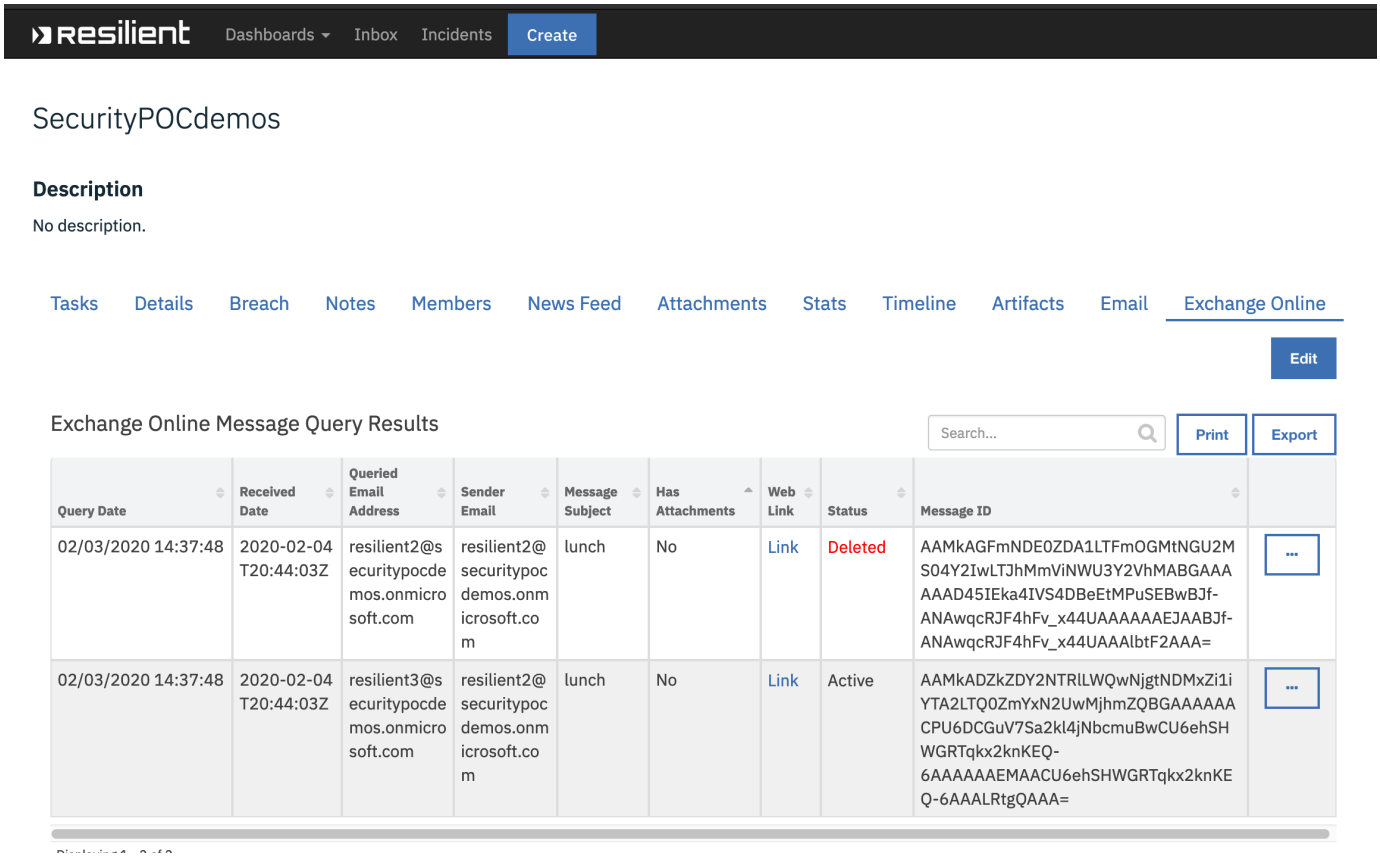
▶ Example Rule:



Data Table - Exchange Online Message Query Results

The following is an example of message query results that are populated in the Exchange Online Message Query Results data table:

NOTE: The Web Link column contains a link to the message, but you must be logged in as the message owner user to view the message in the link.



API Name:

exo_message_query_results_dt

Columns:

Column Name	API Access Name	Type	Tooltip
Queried Email Address	exo_dt_email_address	text	-
Has Attachments	exo_dt_has_attachments	boolean	-
Message ID	exo_dt_message_id	text	-
Message Subject	exo_dt_message_subject	text	-
Query Date	exo_dt_query_date	datetimepicker	-
Received Date	exo_dt_received_date	text	-
Sender Email	exo_dt_sender_email	text	-
Status	exo_dt_status	textarea	-
Web Link	exo_dt_web_link	textarea	-

Rules

Rule Name	Object	Workflow Triggered
Example: Exchange Online Write Message JSON as Note	exo_message_query_results_dt	example_exchange_online_get_message
Example: Exchange Online Send Message	incident	example_exchange_online_send_message
Example: Exchange Online Get User Profile	artifact	example_exchange_online_get_user_profile
Example: Exchange Online Delete Message from Query Results	incident	example_exchange_online_delete_messages_from_query_results

Rule Name	Object	Workflow Triggered
Example: Exchange Online Move Message to Folder	exo_message_query_results_dt	example_exchange_online_move_message_to_folder
Example: Exchange Online Query Messages on Artifact	artifact	example_exchange_online_query_emails
Example: Exchange Online Write Message EML as Attachment	exo_message_query_results_dt	example_exchange_online_write_message_as_attachment
Example: Exchange Online Query Messages	incident	example_exchange_online_query_messages_of_a_group
Example: Exchange Online Create Meeting	incident	example_exchange_online_create_meeting
Example: Exchange Online Create Artifacts	exo_message_query_results_dt	–
Example: Exchange Online Delete Message	exo_message_query_results_dt	example_exchange_online_delete_email